



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**VYUŽITÍ UMĚLÉ INTELIGENCE K MONITOROVÁNÍ
STAVU OBRÁBĚCÍHO STROJE**

USING ARTIFICIAL INTELLIGENCE TO MONITOR THE STATE OF THE MACHINE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Nikola Popara

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Kovář, Ph.D.

BRNO 2020

Zadaní diplomové práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Bc. Nikola Popara**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Jiří Kovář, Ph.D.**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Využití umělé inteligence k monitorování stavu obráběcího stroje

1 Stručná charakteristika problematiky úkolu:

Cílem práce je použití existující platformy obsahující některé metody tzv. umělé inteligence k monitorování stavu stroje vyhodnocováním měřených dat.

2 Cíle diplomové práce:

Osvojte si práci s neuronovými sítěmi a platformou UVSSR Neural Network.
Natrénujte neuronální síť za účelem monitorování technického stavu stroje.
Ověřte navrženou síť v reálných podmínkách.

3 Seznam doporučené literatury:

MITCHELL, T. M., Machine Learning, McGraw-Hill Science/Engineering/Math, 1997, ISBN: 0070428077.

BISHOP Ch. M., Pattern Recognition and Machine Learning, Springer S+M Media, 2006, ISBN: 0-38-31073-8.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Práca je zameraná na monitorovanie najviac namáhaných častí obrábacieho stroja. Použitá metóda umelej inteligencie je rekurentná neurónová sieť a jej modifikácie. Nakoľko dáta zo senzorov mali sekvenčný charakter, bolo vhodné zvoliť práve rekurentnú neurónovú sieť. Práca sa zaoberá riešením troch úloh. Prvá úloha bola zameraná na stanovenie predpokladaného opotrebenia frézy, na základe nepriamej metódy využívajúcej neurónovú sieť. Ďalšia úloha sa zameriava na detekciu poruchy ložiska na základe dát získaných z akcelerometra. Treťou úlohou bolo predikovať dobu do poškodenia monitorovaného ložiska.

Kľúčové slová: rekurentná neurónová sieť, LSTM, GRU, predikcia RUL, monitorovanie ložiska, stav opotrebenia frézy

ABSTRACT

This thesis is focus on monitoring state of machine parts that are under the most stress. Type of artificial intelligence used in this work is recurrent neural network and its modifications. Chosen type of neural network was used because of the sequential character of used data. This thesis is solving three problems. In first problem algorithm is trying to determine state of mill tool wear using recurrent neural network. Used method for monitoring state is indirect. Second Problem was focused on detecting fault of a bearing and classifying it to specific category. In third problem RNN is used to predict RUL of monitored bearing.

Key words: recurrent neural network, LSTM, GRU, prediction of RUL, condition monitoring of a bearing, degradation state of a bearing

BIBLIOGRAFICKÁ CITÁCIA

POPARA, N. *Využití umělé inteligence k monitorování stavu obráběcího stroje*. **Brno:** Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2021. 41 s. Vedúci diplomovej práce Ing. Jiří Kovář, Ph.D..

PREHLÁSENIE

Prehlasujem, že som diplomovú prácu na tému **Využití umělé inteligence k monitorování stavu obráběcího stroje** vypracoval samostatne, s použitím odbornej literatúry a prameňov, uvedených v zozname, ktorý tvorí prílohu tejto práce.

.....
Dátum

.....
Nikola Popara

POĎAKOVANIE

Ďakujem Ing. Jiřímu Kovářovi Ph.D. za cenné pripomienky a rady, ktoré mi poskytol pri vypracovaní záverečnej práce.

Ďalej by som sa rád poďakoval svojej rodine za podporu počas celého štúdia.

Obsah

1	Úvod.....	12
2	Rešerš	13
2.1	Priemysel 4.0.....	13
2.2	Umelá inteligencia a monitorovanie stavu strojov	14
2.2.1	Architektúra neurónových sietí.....	16
2.2.2	Neurónové siete typu RNN (Rekurentné neurónové siete)	17
2.2.3	Neurónové siete typu LSTM (Long short-term memory)	19
2.2.4	Neurónové siete typu GRU (Gated recurent unit)	20
2.2.5	Aktivačná funkcia.....	21
2.2.6	Chybová funkcia.....	22
2.2.7	Koeficient učenia	24
2.2.8	Optimizátory	25
2.2.9	Backpropagation (Spätná propagácia).....	27
2.3	Príprava dát na tréning neurónovej siete.....	29
2.4	Filtrácia dát pomocou Wavelet transformácie.....	30
2.5	RUL	31
3	Praktická časť monitorovania obrábacieho stroja	32
3.1	Základný prehľad použitých algoritmov	32
3.1.1	Algoritmus určovania momentálneho stavu nástroja	33
3.1.2	Algoritmus na detekciu a klasifikáciu poruchy	34
3.1.3	Algoritmus na monitorovanie a predikciu stavu ložiska	35
3.2	Popis riešených problémov v praktickej časti práce	36
3.3	Problém 1 - Predikcia stavu frézy	36
3.3.1	Rozpis datasetu opotrebenia CNC stroja	36
3.3.2	Postup riešenia monitorovania stavu obrábacieho nástroja	40
3.3.3	Výsledky neurónovej siete po tréningu.....	40
3.3.4	Aplikácia na monitorovanie opotrebovania frézy.....	42
3.4	Problém 2 - Detekcia chyby ložiska.....	43
3.4.1	Dataset porúch ložísk.....	43
3.4.2	Postup riešenia detekcie poruchy ložiska	43
3.4.3	Výsledky	44
3.5	Problém 3 - Monitorovanie stavu ložiska.....	45
3.5.1	Dataset priebehu života ložiska	45
3.5.2	Postup riešenia predikcie RUL ložiska.....	47
3.5.3	Výsledky	49
3.5.4	Aplikácia na monitorovanie ložiska	50
4	Zhodnotenie praktickej časti	51
5	Záver	52
6	Zoznam použitých zdrojov.....	53
7	Zoznam použitých symbolov	58

8	Zoznam použitých skratiek.....	59
9	Zoznam použitých obrázkov	60
10	Zoznam použitých tabuliek	61

1 Úvod

V priemysle je dôležité dbať na prevádzkyschopnosť strojov. Odstávka obrábacieho stroja môže zapríčiniť pozastavenie produkcie, a to má za následok ekonomické straty. Z dôvodu zamedzenia ekonomických strát je vhodné monitorovať stroje. Monitorovaním strojov sa predchádza ich poruchám. Na zamedzenie vzniku porúch sa používajú dva typy údržby strojov. Jednou metódou je proaktívna údržba, kde pracovník po uplynutí určitej doby skontroluje požadovaný stroj. Efektívnejšou metódou je použitie prediktívnej údržby. Úlohou prediktívnej údržby je schopnosť predpovedať výskyt poruchy častí strojov, v našom prípade obrábacieho stroja.

Predikcia poruchy obrábacieho stroja sa môže zabezpečiť prostredníctvom neurónových sietí. V súčasnej dobe sú na mnohé aplikácie v priemysle čoraz častejšie využívané neurónové siete. Trendom v priemysle je aplikovať senzory, ktoré sú vhodné na monitorovanie stavu stroja. Vďaka získavaným dátam zo senzorov je možné trénovať neurónové siete na predikciu stavu obrábacieho stroja.

Neurónová sieť je tým algoritmom umelej inteligencie, ktorý je inšpirovaný ľudskými neurónmi. Cieľom umelej neurónovej siete je vytvoriť model neznámeho systému na základe znalosti vstupných a výstupných dát z tohto systému. Neurónová sieť je schopná napodobniť požadovaný systém bez jeho analytickej znalosti. Funguje ako tzv. „black box“.

Táto práca je rozdelená na dve časti. V prvej časti je vypísaný teoretický základ použitých algoritmov. Druhá časť sa zaoberá praktickou realizáciou monitorovania obrábacieho stroja, resp. jeho najviac namáhaných častí. Použitý typ neurónovej siete na monitorovanie stavu časti obrábacieho stroja bola rekurentná neurónová sieť a jej modifikácie. Praktická časť tejto práce je rozdelená na riešenie troch úloh.

Prvou úlohou práce bolo monitorovanie stavu opotrebenia frézy, s použitím dát získavaných zo senzorov. Použitými senzormi boli akcelerometer, dynamometer a akustický senzor. Na základe vstupných dát predpokladala rekurentná neurónová sieť aktuálny stav opotrebenia frézy. Použili sa tri typy rekurentných neurónových sietí a to LSTM, RNN a GRU.

Druhá úloha sa zaoberá klasifikáciou a detekciou poškodenia ložísk. Ložiská sú podstatnou súčasťou každého obrábacieho stroja a sú vysoko namáhanou súčasťou. Z dôvodu veľkej záťaže ložísk sú tieto náchylne na poškodenie. Poškodenie by malo za následok odstávku stroja. Na detekciu chyby ložiska sa zvolila neurónová sieť typu LSTM.

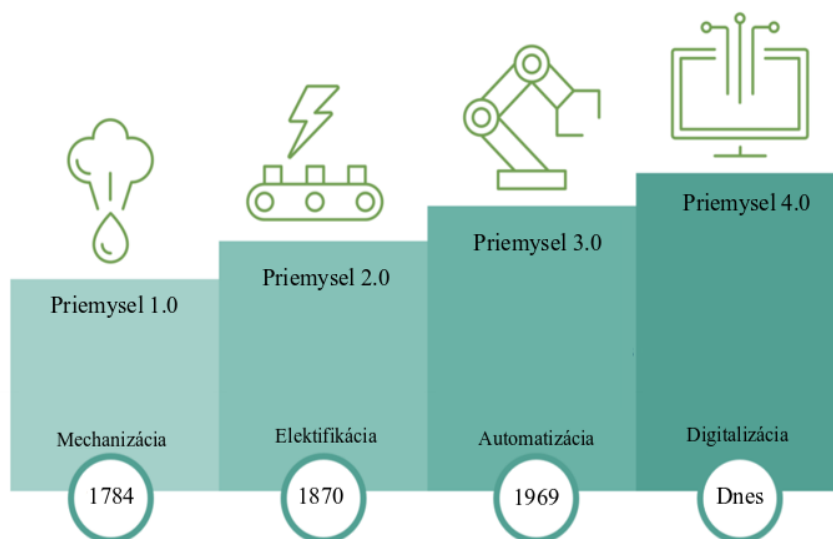
Tretia úloha praktickej časti tejto práce je zameraná na predikciu zostávajúcej doby do poškodenia ložiska. Pomocou akcelerometra sa monitorovali vibrácie ložiska v dvoch osách. Dáta zo senzora boli použité na trénovanie LSTM neurónovej siete. Sieť mala za úlohu predikovať zostávajúcu dobu do doby poškodenia ložiska. Predikcia zostávajúcej životnosti ložiska sa spustila po prekročení počiatočného opotrebenia. Po natrénovaní neurónovej siete bola vytvorená aplikácia v Pythone. Aplikácia mala za úlohu monitorovať stav ložiska, a v prípade detekcie poruchy začať s predikciou životnosti ložiska.

2 Rešerš

Rešeršná časť práce sa zaoberá teoretickým úvodom problematiky neurónových sietí a ich využitia v Priemysle 4.0. Postupne sú uvedené a vysvetlené jednotlivé zložky neurónových sietí. Taktiež je poskytnutý teoretický základ k praktickej časti, ktorá sa zaoberá využitím neurónových sietí k monitorovaniu stavu obrábacieho stroja.

2.1 Priemysel 4.0

Počas histórie sme prešli tromi industriálnymi revolúciami a momentálne sa nachádzame v priemyselnej revolúcii 4.0. Ľudia sa od nepamäti snažili uľahčovať si a zefektívňovať svoju prácu. Pri prvej industriálnej revolúcii sa v priemysle na poháňanie zariadení využila energia vody a vodnej pary. V roku 1870[1] sa urýchlila výroba pomocou montážnych liniek a využitím elektrickej energie. Vďaka tomu sa umožnila aj masovejšia výroba. Vývojom prvých počítačov sa umožnila ešte väčšia automatizácia výroby. V priemysle boli určité pracovné pozície plne nahradzované robotom. V súčasnosti je snaha prepájať všetky automatizované komponenty výroby a umožniť komunikáciu aj na veľké vzdialenosti [2].



Obr. 1: Evolúcia priemyslu [3]

Na fungovanie plne automatizovanej výroby, ktorá je schopná navzájom komunikovať, je potrebných niekoľko dôležitých faktorov. Jednou z nich je výkonná riadiaca jednotka, ktorá je aj cenovo prijateľná. Vo svete je tendencia zvyšovania výkonu a cenovej dostupnosti riadiacích jednotiek [4]. Vďaka ľahšej prístupnosti k výkonným riadiacim jednotkám je umožnené väčšie automatizovanie fabrík. Ďalšou dôležitou zložkou revolúcie 4.0 je zhromažďovanie veľkého množstva dát. Dáta sú potrebné na vyhodnocovanie a pre následnú predikciu činnosti zariadení. Bez informácií by sa nedali riadiť zariadenia v priemysle. Zariadenia musia byť schopné v reálnom čase vyhodnocovať dáta / informácie a preposielať ich na ďalšie vyhodnotenie. Následne je dôležité zabezpečiť rýchlu a efektívnu komunikáciu.

2.2 Umelá inteligencia a monitorovanie stavu strojov

Umelá inteligencia je schopnosť prístrojov vnímať okolie, a na základe vopred daného cieľa hľadá optimálne riešenie problému [5]. Dalo by sa povedať, že sa jedná o snahu napodobniť ľudské správanie. Existuje celý rad algoritmov umelej inteligencie. Jednou podmnožinou umelej inteligencie je strojové učenie alebo Machine Learning (ďalej ML). Jedná sa o algoritmus, ktorý je trénovaný dátami a schopný automaticky sa upravovať na základe predošlých skúseností [6]. Algoritmus ML je schopný riešiť daný problém aj v prípade, že nebol explicitne naprogramovaný na riešenie daného problému. Podľa spôsobu učenia sa delí strojové učenie do dvoch hlavných skupín, buď na trénovanie bez učiteľa alebo na trénovanie s učiteľom.

Trénovanie bez učiteľa je technika na učenie siete, ktorá rozpoznáva vzory na základe vstupných dát. Neurónová sieť nepotrebuje k učeniu vopred známe výstupné hodnoty. Takto učný algoritmus je schopný detegovať chybu bez toho, aby sa vopred vyskytla. Technika učenia siete bez učiteľa sa najčastejšie využíva pre detekciu anomálií, resp. porúch strojov. [7]

Trénovanie s učiteľom je proces učenia, v ktorom k vstupu do modelu neurónovej siete je poskytnutý požadovaný výstup z neurónovej siete. Algoritmus vytvára model, ktorého úlohou je prepojiť poskytnutý vstup k požadovanému výstupu. Trénovanie s učiteľom sa používa na klasifikáciu, na regresnú analýzu a na techniky, ktoré sú schopné vykonávať oboje, teda aj klasifikáciu, aj regresnú analýzu. [7]

Najčastejšie používanou metódou na ML je Deep Learning (ďalej DL). Inšpirácia pochádza z ľudskej neurónovej siete [8]. Snahou je syntetizovať ľudskú inteligenciu. Algoritmus sa skladá z neurónov poprepájaných s váhami. Neurónová sieť obsahuje vstupnú vrstvu, skrytú vrstvu a výstupnú vrstvu. Počet skrytých vrstiev je teoreticky neohraničený. Väčšinou so zvyšovaním počtu vrstiev sa zvyšuje aj presnosť a schopnosť algoritmu riešiť problémy. Nevýhodou a hlavnou limitáciou je výkonnosť súčasných počítačov. Vlastnosti neurónových sietí sú definované mnohými faktormi. Pred spustením učenia neurónovej siete je možné si zvoliť aktivačnú funkciu, chybovú funkciu, počet neurónov, počet vrstiev a typ architektúry neurónovej siete (ďalej architektúry). V prípade nastavenia neurónovej siete sa spustí učenie. Počas učenia si neurónová sieť vhodne nastavuje váhy medzi neurónmi, a tým sa učí. Základné typy úloh pre neurónové siete sú: [9]

- klasifikácia rozdeľuje dáta do niekoľkých tried
- regresia odhaduje číselnú hodnotu vstupu podľa výstupu
- zhľukovanie zaraďuje objekt do skupín s podobnými vlastnosťami

V súčasnej dobe sa neurónové siete uplatňujú na množstvo aplikácií, napr. strojové videnie, automatická premena reči na text, rozpoznávanie hlasu, monitorovanie stavu strojov a procesov. Tradičnou metódou na monitorovanie stavu strojov a procesov je fyzikálna reprezentácia modelu daného stroja alebo procesu. Nevýhodou takéhoto modelu je náročnosť tvorby samotného modelu a jeho následnej úpravy. Týmto spôsobom tvorený model nie je schopný prispôbiť sa novým dátam a novým okolnostiam. Riešením spomenutých problémov je práve vytvorenie daného modelu pomocou neurónovej siete. Proces tvorby modelu je výrazne jednoduchší. V prípade dodatočných zmien je ľahšie upraviť, resp. doučiť model. Na získanie spoľahlivého modelu z neurónových sietí je potrebné disponovať veľkým množstvom dát. Druh dát závisí od aplikácie neurónovej siete. Neurónová sieť na rozpoznávanie obrazov pre svoju funkciu rozpoznávania potrebuje vopred veľké množstvo obrazov, na základe ktorých

sa naučí rozpoznávať dané obrazy. V prípade monitorovania strojov je potrebné dodať dáta z vhodných senzorov. Najčastejšie používané senzory na monitorovanie výrobných strojov sú akcelerometre, dynamometre alebo senzory monitorujúce elektrické napätie a prúd.

Neurónové siete majú niekoľko variant, napr. auto-enkóderi, konvolučné neurónové siete, rekurentné neurónové siete. Každá z nich má pre riešenie rôznych problémov svoje uplatnenie. Konkrétnejšie vysvetlenie jednotlivých typov neurónových sietí je v kapitole 2.2.1.

Napríklad pán G. Tang použil v práci [10] na predikciu degradácie ložiska kombináciu auto-enkóderu a LSTM (*Long short-term memory*). Cieľom bolo získať vhodné reprezentačné dáta, ktoré boli následne použité na tréning LSTM siete. Výstupom z neurónovej siete bola absolútna hodnota aritmetického priemeru chyby. Neurónová sieť bola schopná predikovať priebeh stavu zaťažovaného ložiska.

Ďalšia práca využívajúca LSTM na monitorovanie bola od M. Yuana. Snahou uvedeného algoritmu je predikovať ostávajúcu životnosť turbín s použitím LSTM metódy [11]

V práci od R. Liu [12] je uvedená dislokovaná časová séria konvolučnej neurónovej siete (DTS-CNN), ktorú použil na detekciu porúch elektrického stroja. Snahou je odstrániť nedostatky CNN v aplikácii monitorovania elektrických strojov. Štruktúra siete DTS-CNN sa skladá z dislokovanej vrstvy, konvolučnej vrstvy a plne prepojenej vrstvy. Výhodou pridania dislokačnej vrstvy je extrahovanie vzťahov medzi signálmi s rôznymi intervalmi, v periodicky sa opakujúcich mechanických signáloch.

V článku od E. Sutrisného [13] boli použité tri metódy na predikciu životnosti ložiska. Prvá metóda použila spektrálny koeficient špicatosti signálu zrýchlenia na predikciu ostávajúcej životnosti ložiska. Hodnota zrýchlenia sa získavala z akcelerometrov, ktoré boli umiestnené na monitorovanom ložisku. Dôvodom výberu koeficientu špicatosti na predikciu stavu ložiska bol v časovej oblasti jeho monotónny trend. Táto pozorovaná veličina bola vhodná na monitorovanie opotrebenia ložiska na dlhšom časovom zázname. Na predikciu životnosti ložiska sa využíval exponenciálny model, ktorý nebol schopný na konci jeho životnosti zachytiť náhlu zmenu opotrebenia ložiska. Hlavným problémom danej metódy bolo preceňovanie doby životnosti pozorovaného ložiska.

V druhej metóde na rozdiel od prvej metódy nebolo potrebné použiť reprezentáciu signálu, ktorá má monotónny charakter. Použitá technika bola schopná rozoznať o aký trend sa jedná a na základe toho predikovala životnosť ložiska. Nakoľko všetkých 6 ložísk, na ktorých bola metóda trénovaná, mali rozdielnú dobu životnosti, daná metóda nebola schopná správne predikovať životnosť ložiska. Daná metóda by potrebovala väčšie množstvo dát. V prípade poskytnutia väčšieho datasetu by sa mohla výrazne zvýšiť presnosť metódy predikovať zostávajúcu životnosť ložiska. [13]

Tretia prezentovaná metóda porovnávala dĺžku degradačného úseku trénovaných dát a testovacích dát. Pri tvorbe modelu boli vylúčené dva záznamy z opotrebenia ložiska, nakoľko bola zistená anomália vo frekvenčnom pásme. Po odstránení týchto dvoch záznamov opotrebenia ložísk bola metóda schopná predikovať životnosť ložiska. [13]

V práci od Y. Zhanga a A. Wang [14] je použitá modifikácia stochastickej filtrácie na predikciu životnosti ložiska. Dáta z opotrebenia ložiska sú rozdelené na dva celky. V časti reprezentujúcej normálny stav ložiska sa monitoruje len stav ložiska. Naopak v degradačnej oblasti záznamu z opotrebenia ložiska sa používa algoritmus na predikciu životnosti monitorovaného ložiska.

2.2.1 Architektúra neurónových sietí

Existuje veľké množstvo variant neurónových sietí. Jednotlivé architektúry majú svoje prednosti a používajú sa na rozdielne účely. K dispozícii je sortiment neurónových sietí, ktorý sa neustále zväčšuje. Vo väčšine prípadov sa dá vybrať z už existujúcich architektúr. V prípade, ak sa jedná o špecifický problém, pristupuje sa k navrhovaniu vlastnej architektúry.

Najjednoduchšou architektúrou je jeden neurón, s jedným vstupom a s jedným výstupom. V takomto prípade je funkcia tejto siete limitovaná. Prakticky je nevyužiteľná. Ďalej sa dajú rôznymi spôsobmi vytvárať zložitejšie štruktúry, ktoré sa skladajú z mnohých neurónov a môžu byť poprepájané niekoľkými spôsobmi. V tejto kapitole si vymenujme niekoľko zaujímavých príkladov neurónových sietí.

FFNN (*Feedforward neural network*) je základný typ neurónovej siete. Informácia prechádza niekoľkými neurónmi zo vstupu až na výstup. Jednotlivé neuróny sú navzájom paralelné a zoskupené do vrstiev. Pri dostatočnom počte neurónových vrstiev je sieť schopná nadobudnúť vzťah k ľubovoľnému vstupu a výstupu. [15]

RNN (*Recurrent neural network*) je neurónová sieť, ktorej neuróny majú dva zdroje informácií. Do neurónu vstupuje informácia z predošlého neurónu a ešte uchovaná informácia z minulého učenia. Neurón si uchováva históriu a sám si ju vkladá ako vstup ku ďalšiemu časovému kroku. Táto vlastnosť sa využíva pri sekvenčných dátach. Jedným z problémov tejto siete je strácanie gradientu, a to znemožňuje prácu s časovými dátami, ako je zvuk alebo video [16]

AE (*Auto-encoder*) sa skladá z dvoch FNN sietí. Výstup prvej siete slúži ako vstup druhej siete. Tvar architektúry AE pripomína presýpacie hodiny. Druhá sieť je zrkadlový obraz prvej siete. Pri učení je vstup do siete rovnaký ako výstup. Proces sa skladá z kódovania a dekódovania. V procese enkódera sa zo vstupu redukuje dimenzia dát. Ďalej dekóder slúži na rekonštrukciu pôvodnej dimenzie dát. [17]

DBN (*Deep belief network*) sa skladá z niekoľkých, po sebe idúcich VAE (*Variational autoencoder*) sietí. Základ VAE algoritmu spočíva v distribúcii latentných premenných. K učeniu latentnej reprezentácie sa používa variačná zložka. Umožňuje to zhromaždenie naučených množín do jednej oblasti tak, aby jednotlivé množiny boli čo najbližšie k sebe. U latentných vektorov sa predpokladá Gaussova distribúcia [17]. Každá sekcia si hľadá svoje optimálne riešenie. Tým pádom sieť získava lokálne optimá. [18]

CNN (*Convolutional neural network*), jedná sa o typ neurónovej siete, ktorá sa líši oproti klasickým neurónovým sieťam lokálnou konektivitou neurónov a zdieľaním parametrov s ostatnými neurónmi [19]. Najčastejšie použitie CNN je v klasifikácii obrazov.

Tab. 1: Porovnanie výhod a nevýhod jednotlivých architektúr neurónových sietí [20]

Algoritmus	Výhody	Nevýhody
CNN	Analýzu multi-dimenzionálnych dát Extrahovanie lokálnych prvkov	Komplikovaná štruktúra siete Vyžaduje dlhú dobu na výpočty
RNN	Práca so sekvenčnými dátami Využitie dát z časovej oblasti	Náročné na implementáciu a tréningovanie
DBN	Analýza jednodimenzionálnych dát Extrakcia globálnych prvkov z neupraveného záznamu	Neefektívne a pomalé
AE	Ľahké na implementáciu	Potreba veľkého množstva dát

2.2.2 Neurónové siete typu RNN (Rekurentné neurónové siete)

Jedná sa o typ neurónovej siete, kde výstup z predošlého časového kroku slúži ako vstup súčasného časového kroku. Rekurentná neurónová sieť je vhodná pri práci s dátami, ktoré majú sekvenčný charakter. Dáta, ktoré sú závislé na poradí, v ktorom po sebe nasledujú, sa označujú ako sekvenčné dáta [21]. Príkladom sekvenčných dát je napr. audiozáznam alebo hodnoty zo senzorov ako je akcelerometer alebo dynamometer.

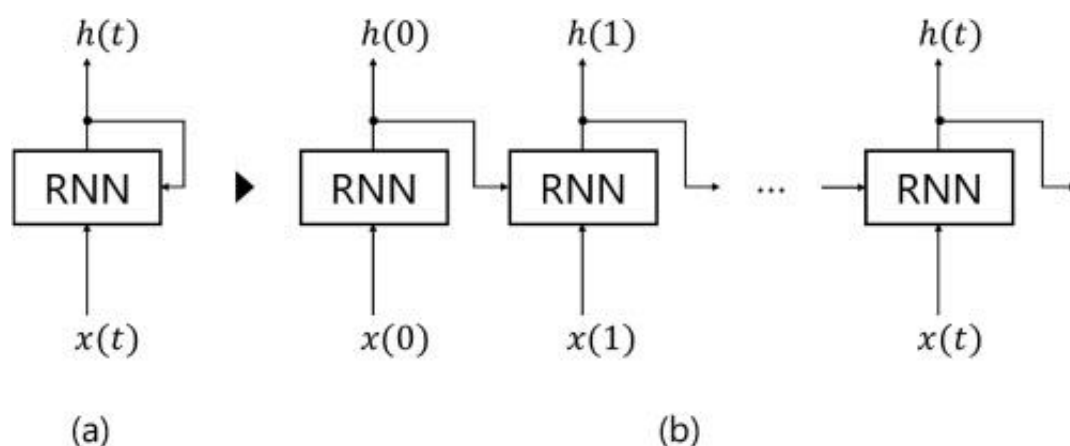
Na efektívnu prácu so sekvenčnými dátami bola vytvorená RNN. V článku od Schimdhuberta bolo spomenuté, že sa jedná o najhlbší typ neurónovej siete [22]. Neurónová sieť je schopná zachytávať všetky predošlé vstupy do vnútornej pamäte a priradovať k nim cieľový výstup [23]. Táto vlastnosť umožňuje ukladať do vnútornej pamäte všetky predošlé vstupy. Vďaka pamäti je umožnené zachytávanie vzťahov medzi jednotlivými vstupnými hodnotami. Inými slovami, sieť je schopná vykonávať predikciu hodnoty na základe detegovanej sekvencie vstupov.

Vnútna pamäť v RNN umožňuje prácu so sekvenčnými dátami na vstupe. Prechodová funkcia v každom časovom kroku zabezpečuje prepojenie aktuálneho vstupu x_t s predošlým vnútorným stavom siete h_{t-1} . Rovnica pre výpočet výstupu vnútorného stavu v čase t je:

$$s_t = Wx_t + Vh_{t-1} + b \quad (1)$$

$$h_t = f(s_t) \quad (2)$$

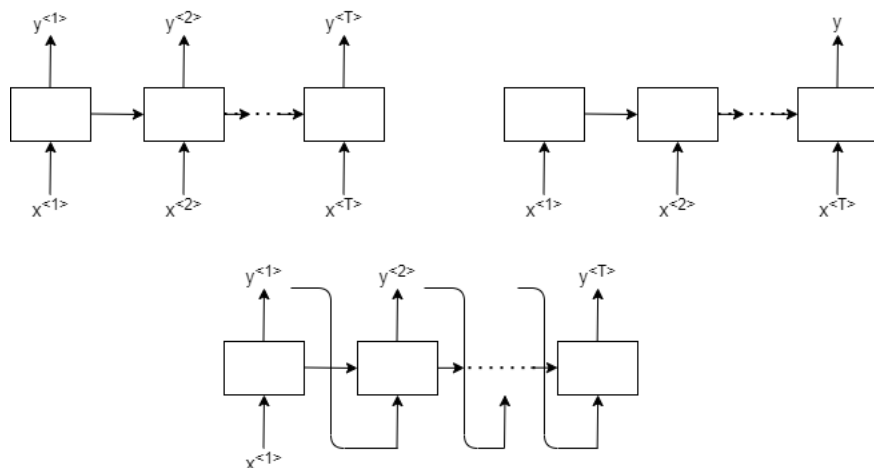
W a V sú matice váh, kde W je matica váh pre hodnoty vstupov a V je matica pre váhy skrytej vrstvy h_{t-1} . Tretou zložkou b je vektor biasa. Rovnica obsahuje aktivačnú funkciu $f(s_t)$, kde f je nelineárna funkcia, najčastejšie hyperbolický tangens. Po spracovaní celej sekvencie je výstup z vnútorného stavu siete h_t naučenou reprezentáciou sekvencie vstupov, ktorej dĺžka je T . [23]



Obr. 2: Rozbalenie rekurentnej neurónovej siete [24]

Rekurentné siete na základe pomeru vstupov a výstupov by sa dali kategorizovať do troch skupín: [25]

- rekurentná sieť s výstupom v každom čase a spätným prepojením skrytých jednotiek
- rekurentná sieť s výstupom v každom čase, kde výstup siete je prepojený so skrytými jednotkami
- rekurentná sieť, ktorá má prepojenia medzi skrytými jednotkami, spracuje kompletnú sekvenciu a produkuje len jednu výstupnú hodnotu



Obr. 3: Rozdelenie RNN podľa pomeru vstupov a výstupov z neuónovej siete

Jedným z ďalších možných spôsobov kategorizovania RNN je na základe počtu vrstiev:

- S jednou vrstvou, tzv. jednovrstvové Vanilla RNN. Jedná sa o najjednoduchší typ rekurrentnej neurónovej siete.
- Rekurentné neurónové siete s niekoľkými vrstvami. Na zefektívnenie nelineárnych operácií zo vstupných dát je vhodné použiť niekoľkovrstvovú RNN. Dôvodom zefektívnenia je fakt, že je umožnené distribuovať parametre v priestore, a tým zmenšiť potrebnú veľkosť pamäte. Viacvrstvomá sieť umožňuje propagáciu vnútorného stavu v čase a sprostredkováva ho ako vstup pre ďalšiu vrstvu [26].
- Ďalším riešením je obojsmerná RNN (bi-directional RNN alebo BiRNN). Rozdielom BiRNN a RNN je v jej schopnosti spracovávať sekvenčné dáta v oboch smeroch a má dve samostatné skryté vrstvy.

Rekurentné neurónové siete môžeme použiť na niekoľko účelov, ktoré využívajú dáta v časovej oblasti. Kategoricky delíme rekurentné neurónové siete podľa použitia na: [27]

- klasifikácia s časovým kontextom
- predikcia časovej postupnosti
- modelovania časovej postupnosti

Na odstránenie niektorých nedostatkov jednovrstvovej Vanilla RNN boli vyvinuté rôzne modifikácie, ako napr. LSTM a GRU. O modifikáciách LSTM a GRU sa bude táto práca ďalej zaoberať v kapitolách 2.2.3 a 2.2.4.

2.2.3 Neurónové siete typu LSTM (Long short-term memory)

V prípade jednoduchého RNN existujú dva zásadné problémy, ktoré sa vyskytujú pri učení siete počas jej spätnej propagácie. Jeden z problémov je propagácia chyby v čase počas procesu učenia, kde gradient môže dosiahnuť hodnoty blízku nule. Sieť nie je schopná efektívne upravovať váhy, nakoľko hodnota gradientu je výrazne menšia. Druhým problémom je tzv. explodovací gradient (exploding gradient). V prípade, keď algoritmus priradí nesprávne a prehnane veľkú vážnosť niektorej z váh siete. V roku 1997 bola pánmi Hochreiterom a Schmidhuberom vylepšená RNN tak, aby odstránila vyššie spomenuté problémy s miznúcim gradientom [28]. Vylepšená metóda RNN sa označuje ako LSTM (*Long short-term memory*). Princíp spomenutej metódy spočíva v kontrole toku dát pomocou tzv. brán (gates). Tieto brány slúžia ako dodatočná regulácia vstupov. V každom časovom kroku sa využíva niekoľko brán, ktoré sú schopné efektívne riadiť pohyb dát v jednotlivých blokoch. Každý z blokov v LSTM môžeme rozdeliť na nasledujúce časti:

Stav pamäťového bloku (*cell state*) sa označuje c_t . Každý blok LSTM siete obsahuje kanál c_t , v ktorom sa upravujú a pridávajú informácie. Kanál zabezpečuje schopnosť siete dlhodobo uchovávať informácie. Výstupom z LSTM bloku je skrytý stav h_t v čase t . Výstupná hodnota h_t je násobkom výstupnej brány a hyperbolického tangensu z c_t . [29]

$$h_t = o_t \cdot \tanh(c_t) \quad (3)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c x_t + V_c h_{t-1} + b_c) \quad (4)$$

Zabúdajúca brána (*forget gate*) má na vstupe hodnotu x_t a výstup z predošlého časového kroku skrytej vrstvy h_{t-1} . Prvým krokom je výpočet dôležitosti pomocou sigmoidnej aktivačnej funkcie. Ďalším krokom je násobenie výsledného vektora so stavom kanálu c_{t-1} . [29]

$$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f) \quad (4)$$

Vstupná brána (*input gate*) umožňuje vstupným dátam x_t zmeniť vnútorný stav. Brána rozhoduje, koľko nových informácií sa pridá do súčasnej pamäte kanálu c_t . [29]

$$i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i) \quad (5)$$

Výstupná brána (*output gate*) určuje množstvo informácií, ktoré sa dostane na výstup z jedného LSTM bloku. [29]

$$o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o) \quad (6)$$

2.2.4 Neurónové siete typu GRU (Gated recurent unit)

Ďalšia modifikácia prišla v roku 2014 vo forme GRU [21]. Blok GRU je podobný bloku LSTM. Obidva tieto algoritmy sú komplexnejšou variantou v porovnaní s Vanilla RNN. Prostredníctvom výstupnej brány (*output gate*) sa v LSTM bloku kontroluje prístup ďalších blokov k informáciám v pamäti. Naopak GRU sprístupňuje celú informáciu z pamäti [28]. Ďalším rozdielom je fakt, že spája f_t a i_t do jednej brány z_t . Nižšie na obrázku sú znázornené dva bloky. Jeden blok znázorňuje LSTM a druhý blok GRU. Každý z blokov v GRU môžeme rozdeliť na nasledujúce časti:

Aktualizujúca brána (*update gate*) rozhoduje o množstve informácií, ktoré sa obnovia v danom bloku [21].

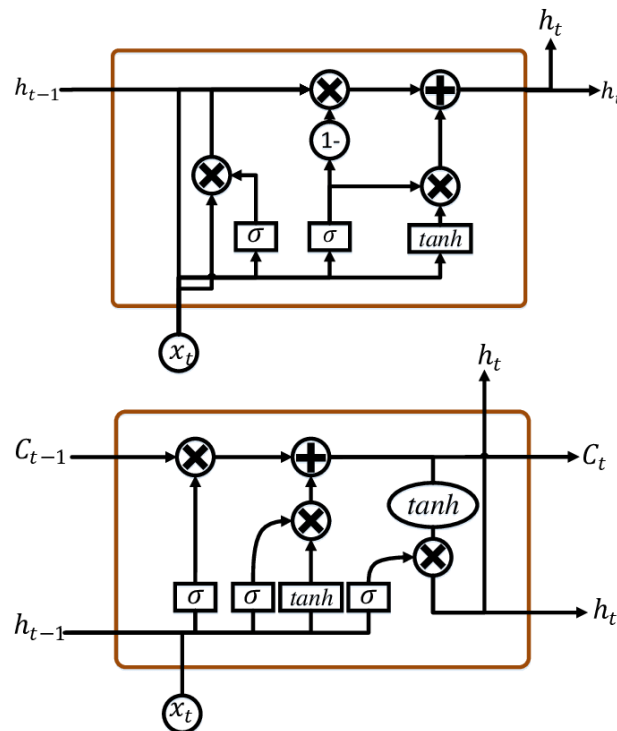
$$z_t = \sigma(W_z x_t + V_z h_{t-1}) \quad (7)$$

Resetovacia brána (*reset gate*) určuje, ktoré informácie sa vymažú. Reset gate umožňuje každému bloku GRU správať sa k vstupu, ako keby ho videl po prvýkrát. Prostredníctvom reset gate sa stratia predošlé informácie [21].

$$r_t = \sigma(W_r x_t + V_r h_{t-1}) \quad (8)$$

Výpočet skrytého stavu, pri ktorom ide o výpočet h_t pomocou lineárnej interpolácie medzi minulým h_{t-1} a hyperbolickým tangensom v rovnici 10 [28].

$$h_t = (1 - z_t)h_{t-1} + z_t \tanh(W x_t + V(r_t \cdot h_{t-1})) \quad (9)$$



Obr. 4: Schematické porovnanie architektúry GRU a LSTM [30]

2.2.5 Aktivačná funkcia

Neurónová sieť sa skladá z jednotlivých neurónov. Každý neurón uskutočňuje jednoduché výpočty prostredníctvom sumácie a aktivačnej funkcie. Aktivačná funkcia definuje výstup z daného neurónu na základe vstupu prvkov do neurónovej siete. Existuje veľa rôznych druhov aktivačných funkcií. Jednotlivé aktivačné funkcie majú svoje výhody a uplatnenia. Najčastejšie sa hovorí o sigmoide. Patrí k jednoduchým a efektívnym aktivačným funkciám. V prípade sigmoidu je aj veľa limitácií a preto sa začali využívať nové typy aktivačných funkcií. [31]

Kroková funkcia patrí k najjednoduchším aktivačným funkciám. Ide o jednoduché rozhodovanie medzi 1 a 0. Nie je vhodné na určovanie pravdepodobnosti výskytu daného objektu, nakoľko neexistuje hodnota medzi nulou a jednotkou.

$$f(x) = \begin{cases} 1 & \text{pre } x > 0 \\ 0 & \text{pre } x < 0 \end{cases} \quad (101)$$

Sigmoidná funkcia je pomerne jednoduchá. Taktiež má podobné charakteristiky ako kroková funkcia. Hlavným rozdielom je schopnosť reprezentovať hodnoty medzi 0 a 1. Dobré sa dá uplatniť pri klasifikácii. Určuje pravdepodobnosť, že daný objekt patrí do danej skupiny.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

Tanh je podobný ako sigmoidná aktivačná funkcia. Na rozdiel od sigmoidnej funkcia hyperbolického tangensu priradzuje k vstupom hodnotu výstupu medzi -1 a 1. Dáta sú viac koncentrované a tým pádom sa neurónová sieť ľahšie učí. Hlavne sa používa v skrytých vrstvách.

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (13)$$

ReLU momentálne patrí k najobľúbenejším aktivačným funkciám. Filtruje záporné hodnoty vstupu. Kladné hodnoty vstupu ponecháva nezmenené. Je veľmi rýchlou metódou na počítanie aktivačnej funkcie a v prípade, ak má neurónová sieť veľa vrstiev, tak je schopná učiť sa komplexné úlohy. Z dôvodu neprítomnosti horného limitu pri ReLU funkcií hrozí nekontrolovaný nárast aktivácie. Na odstránenie niektorých nedostatkov Relu sa používajú funkcie ako je napr. Leaky ReLU. Jedná sa o modifikáciu aktivačnej funkcie ReLU, ktorej záporné hodnoty sa zachovávajú do obmedzenej mier. [31]

$$f(x) = \begin{cases} x & \text{pre } x > 0 \\ 0 & \text{pre } x < 0 \end{cases} \quad (14)$$

Softmax sa používa na binárnu klasifikáciu. Hlavne sa používa na výstupných neurónoch, ktorých úlohou je klasifikácia vstupov do neurónovej siete.

$$f(x) = \frac{e^x}{\sum_x e^x} \quad (15)$$

2.2.6 Chybová funkcia

Slúži na definovanie toho, ako dobre funguje neurónová sieť. Porovnáva výstup z neurónovej siete s vopred známou a vopred požadovanou hodnotou. Jedná sa o jednu hodnotu a nie o vektor. Určuje správnosť výstupu zo siete, ako celku. Vstupná hodnota pre chybovú funkciu je výstup z aktivačnej funkcie s vopred známou hodnotou požadovaného výsledku. Každá chybová funkcia musí spĺňať dve základné podmienky. Prvou podmienkou musí byť možnosť interpretovať chybovú funkciu ako priemer jednotlivých chybových funkcií, z jednotlivých tréningov (epoch). Dôvodom je vypočítanie gradientu po každej jednej epoche. Druhou podmienkou je, aby bola závislá len na aktivačnej hodnote z aktuálneho neurónu a nie na zvyšných neurónoch. Dôvodom je, aby bola umožnená spätná propagácia [32]. Chybové funkcie delíme podľa účelu použitia neurónovej siete na regresné a klasifikačné.

Regresná analýza

Regresná analýza je označenie štatistickej metódy, podľa ktorej sa odhadujú dané veličiny [33]. V neurónových sieťach sa na aproximáciu funkcií používa regresná analýza. V tejto práci sa využije regresná analýza na odhad životnosti ložiska a frézy. Ďalej sú spomenuté tri druhy chybových funkcií, ktoré sa používajú na regresnú analýzu v neurónových sieťach.

MAE (*Mean absolute error*) resp. absolútna priemerná chyba počíta absolútnu hodnotu rozdielu žiadanej hodnoty a hodnoty na výstupe. Počas učenia neurónovej siete s učiteľom je potrebné zistiť, ako blízko je výsledok k požadovanej hodnote. Respektíve, hľadáme vzdialenosť od správneho výsledku. Vzdialenosť sa dá určiť ako rozdiel hodnôt výstupu a požadovanej hodnoty. Keďže nás zaujíma vzdialenosť a nie smer od správnej hodnoty, v tom prípade môžeme uvažovať o absolútnej hodnote rozdielu. Nevýhodou metódy je komplikovanejšie formulovanie gradientu, nakoľko v nule sa nachádza ostrý prechod:

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (16)$$

kde y_i je skutočná hodnota, \hat{y}_i je hodnota predikcie a $|y_i - \hat{y}_i|$ je aritmetický priemer absolútnej chyby.

MSE (*Mean squared error*) resp. stredná kvadratická chyba určuje chybu umocnením rozdielu hodnoty na výstupe s požadovanou hodnotou. Metóda je podobná s metódou MAE. Nevýhodou je citlivosť na extrémny, resp. anomálie [34]:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (17)$$

kde y_i je skutočná hodnota, \hat{y}_i je hodnota predikcie a $(y_i - \hat{y}_i)^2$ je aritmetický priemer druhej mocniny chyby.

Hubertova metóda je kompromisom medzi MAE a MSE. V porovnaní s predchádzajúcimi algoritmami ide o komplexnejšiu metódu. Hubertova metóda je zložená z dvoch častí. V prípade, ak je chyba malá, tak sa správa ako MSE. Naopak, pri výskyte anomálií, resp. väčších chýb sa správa ako MAE: [35]

$$HU = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & ; |y - \hat{y}| \leq \alpha \\ \alpha|y - \hat{y}| & ; \text{ostatne} \end{cases} \quad (18)$$

Klasifikácia

Neurónové siete sa využívajú aj na klasifikáciu vstupov do požadovaných tried. Podľa počtu klasifikácií sa delia na binárne klasifikácie a viacprvkové klasifikácie. V prípade, ak sieť má klasifikovať hodnotu medzi dvomi možnými triedami, tak sa používa binárna klasifikácia. Naopak, ak je viac tried klasifikácie ako dve, tak sa používa viacprvková klasifikácia.

Binárna krížová entropia meria schopnosť neurónovej siete klasifikovať výstupné hodnoty z neurónov, ktorých hodnota je pravdepodobnosť medzi 0 a 1 [36]. Výstup z neurónovej siete je potrebné normalizovať na hodnotu pravdepodobnosti. Výstup zo sigmoidu je 0 až 1. Súčet jednotlivých výstupov zo sigmoidu nedáva hodnotu jedna. Nakoľko pracujeme s pravdepodobnosťou, je potrebné použiť funkciu, ktorá bude normalizovať výstupi zo sigmoidov. Používaná funkcia sa volá softmax: [37]

$$H = -\frac{1}{N} \sum_{i=1}^{N=2} y_i \log(p(y_i)) \quad (19)$$

$$H = -y_i \log(p(y_i)) - (1 - y_i) \cdot \log(1 - p(y_i)) \quad (20)$$

kde y je očakávaný výstup zo siete a $p(y)$, je predpovedaná pravdepodobnosť výskytu očakávaného výstupu v jednej z klasifikačných tried.

Kategorická krížová entropia je schopná klasifikovať požadované hodnoty do 3 a viac klasifikačných tried. Na rozdiel od binárnej krížovej cesty využíva ako aktivačnú funkciu softmax. Rovnica je uvedená nižšie: [37]

$$H = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) \quad (21)$$

$$f(y)_i = \frac{e^{y_i}}{\sum_j^N e^{y_j}} \quad (22)$$

Kombináciou rovnice (11) s rovnicou (12) softmax aktivačnej funkcie, dostaneme upravenú rovnicu pre kategorickú krížovú entropickú chybovú funkciu:

$$H = -\log \left(\frac{e^{y_p}}{\sum_j^N e^{y_j}} \right) \quad (23)$$

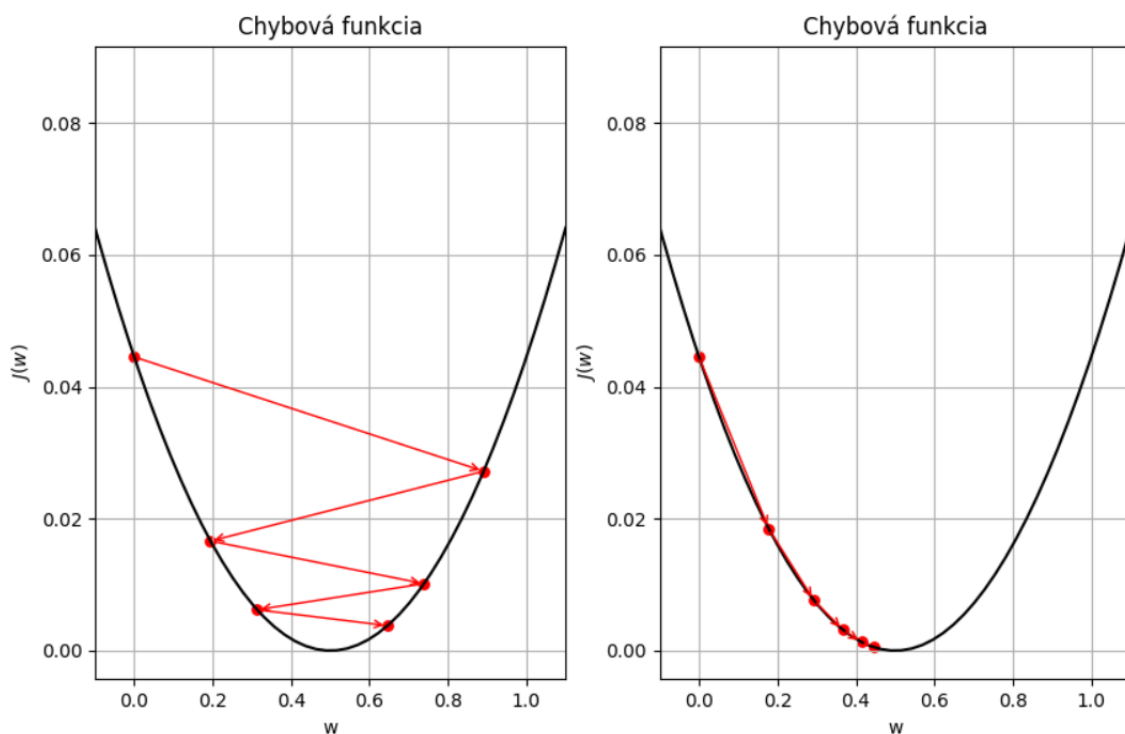
2.2.7 Koeficient učenia

Cieľom učenia neurónovej siete je nájsť minimálnu chybu medzi hľadanou hodnotou a skutočným výstupom z neurónovej siete. Toto učenie sa vykonáva pomocou gradientu, ktorý hľadá minimum danej funkcie. Učenie neurónovej siete je vo svojej podstate len hľadanie minima určitej funkcie. Neurónová sieť si prostredníctvom rovnice upravuje iteratívne váhy:

$$w_{t+1} = w_t - \alpha \nabla J(w) \quad (24)$$

kde w_{t+1} je upravená váha neurónovej siete, w_t je predošlá váha neurónovej siete, α je koeficient učenia a $\nabla J(w)$, vyjadruje Jacobiho maticu vzhľadom k váhe neurónovej siete.

V rovnici figuruje pôvodná váha, od ktorej sa odčíta získaný gradient vynásobený α . Daná konštanta α reprezentuje rýchlosť učenia. Koeficient učenia je pomyselný krok, ktorý vykoná neurónová sieť po jednej iterácii smerom k minimu.



Obr. 5: Porovnanie dvoch hodnôt koeficientov učenia a ich vplyv na konvergenciu

Na obr. 5 je vidno, že je potrebných menej iterácií v prípade zvolenia veľkého kroku Alfa. Neurónová sieť je mnohonásobne rýchlejšia. Problémom veľkého kroku je možnosť prekmitu globálneho minima [38]. Riešením problému s prekmitom minima funkcie je zmenšenie kroku na takú veľkosť, aby sa nemohlo stať to, že sa nezachytí minimum funkcie. V prípade malých krokov je problém s náročnosťou výpočtu. Neurónová sieť potrebuje viac iterácií, aby sa dopracovala k správnejmu výsledku. Hodnota koeficientu učenia sa pohybuje väčšinou medzi hodnotou 0,01 a 0,0001 [39].

2.2.8 Optimizátory

Optimalizácia v matematike sa zaoberá hľadaním minima alebo maxima danej funkcie. V neurónových sieťach sa optimalizáciou priamo neupravujú vstupy. Na základe chybovej funkcie sa upravujú váhy medzi neurónmi. Takto je sieť schopná učiť sa. V neurónových sieťach existuje veľké množstvo používaných optimalizačných metód. Základom týchto optimalizačných metód je gradient poklesu (záporný gradient).

GD (*Gradient descent*) je základom všetkých ostatných metód. Jedná sa o deriváciu chybovej funkcie, ktorá sa následne násobí veľkosťou kroku a odčíta sa od pôvodnej váhy. Nevýhodou je nutnosť počítania gradientu na celom súbore dát. V prípade veľkého súboru dát sa počíta v dlhom časovom intervale:

$$w_{t+1} = w_t - \alpha \nabla J(w) \quad (25)$$

kde w_t je predošlá váha siete, w_{t+1} je nová váha po optimalizácii, α reprezentuje koeficient učenia. Koeficient učenia sa násobí s Jacobiánom danej váhy $\nabla J(w)$.

SGD (*Stochastic gradient descent*) počíta gradient po jednotlivých dátach. Je umožnené rýchlejšie počítanie, resp. rýchlejšie učenie neurónovej siete. Problémom pri SGD je možnosť nedosiahnutia minima, z dôvodu prekmitu. Výhodou je schopnosť nájsť lepšie riešenie, resp. lepšie lokálne minimum.

$$w_{t+1} = w_t - \alpha \nabla J(w; \text{vzorka}) \quad (26)$$

Mini-Batch GD je kompromisom medzi počítaním gradientu na celom súbore dát alebo jednotlivých dátach. Metóda spočíva v rozdelení celkového súboru dát na niekoľko menších podmnožín. Výhodou je zrýchlenie výpočtu a taktiež menšia pravdepodobnosť uviaznutia v lokálnom minime. [40]

$$w_{t+1} = w_t - \alpha \nabla J(w; N \text{ vzorka}) \quad (27)$$

SGD+Momentum slúži ako vylepšenie predošlých metód. Minimalizuje oscilácie počas hľadania lokálneho minima. Ďalšou výhodou je urýchľovanie konvergencie k hľadanému minimu. Rovnice z predošlých príkladov sa upravujú o hodnotu rýchlosti. Dôležitým komponentom je smer z predošlého výpočtu, ktorý sa pridáva do súčasného výpočtu rýchlosti. Vďaka tejto zložke je algoritmus schopný orientovať sa na prevažujúci smer a tým pádom sa zvýši rýchlosť. [41]

$$v_t = \gamma v_{t-1} + \alpha \nabla J(w) \quad (28)$$

$$w_{t+1} = w_t - v_t \quad (29)$$

kde v_t je vektor, ktorý upravuje váhy, γ označuje stupeň priepustnosti hodnôt z predošlého vektora v_{t-1} .

SGD+Momentum+Acceleration najprv sa vykoná interakčný krok na základe predošlej hybnosti. Následne sa vypočíta gradient a potom sa pričíta korekcia. Problém pri predošlej metóde je oscilácia pri ciele. Vďaka tomu, že gradient sa počíta pre ďalší krok, je možné lepšie predpovedať čo sa udeje. [41]

$$v_t = \gamma v_{t-1} + \alpha \nabla J(w - \gamma v_{t-1}) \quad (30)$$

$$w_{t+1} = w_t - v_t \quad (31)$$

Adagrad patrí do súboru adaptívnych optimizátorov. Prednosťou Adagradu je schopnosť adaptívne meniť veľkosť kroku. Najprv sa inicializuje veľkosť kroku ako ľubovoľná vstupná hodnota. Následne sa delí vstupná veľkosť kroku odmocninou G a ε [41]. G sa počíta ako suma všetkých predošlých derivácií chybovej funkcie podľa váhy. V prípade narastania G sa celková veľkosť kroku znižuje:

$$w_{t+1,i} = w_{t,i} - \frac{\alpha}{\sqrt{G_t + \varepsilon}} \nabla J(w_{t,i}) \quad (32)$$

kde $G_t \in R^{d \times d}$ je diagonálna matica, ktorej každý diagonálny prvok je sumou druhej mocniny gradientov. Prvok ε vyjadruje vyrovnávaciu hodnotu, ktorá zabraňuje deleniu nulou.

Adadelta bola navrhnutá tak, aby odstránila nedostatky Adagradu. V prípade hlbokých neurónových sietí sa stáva, že veľkosť kroku je veľmi malá. Spôsobené je to zväčšovaním sa konštanty G . Z toho vychádzajúc je potrebné limitovať minimálnu veľkosť kroku. Rovnica Adadelta sa líši od Adagradu tým, že priamo nepočíta suma gradientov, ale používa priemernú hodnotu. Veľkosť gradientu je limitovaná pomocou $(1 - \gamma)$: [41]

$$w_{t+1,i} = w_{t,i} - \frac{\alpha}{\sqrt{E[g^2]_t + \varepsilon}} \nabla J(w_{t,i}) \quad (33)$$

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) g_t^2 \quad (34)$$

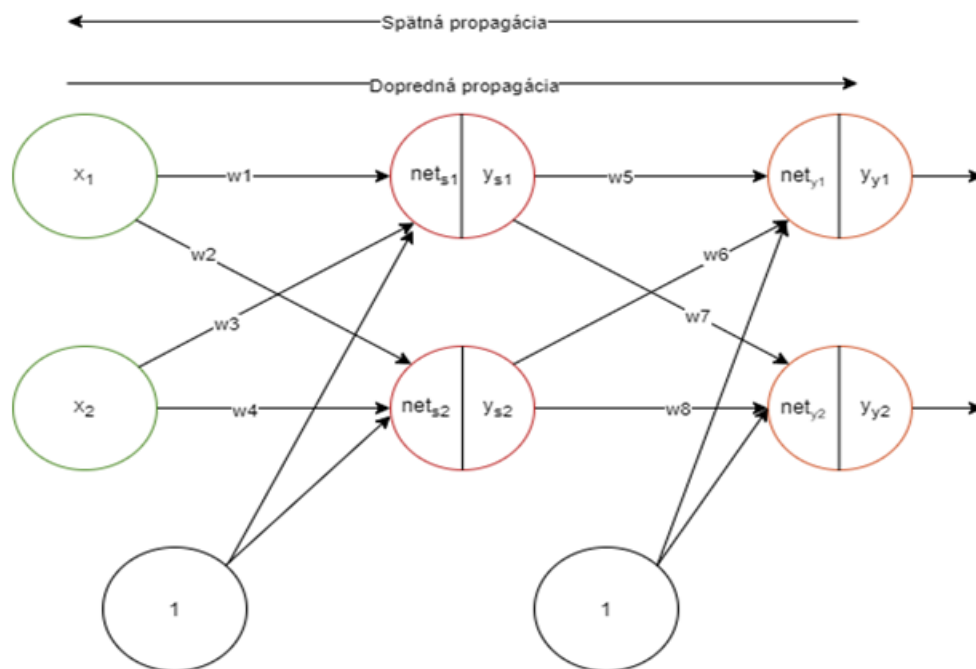
kde $E[g^2]_t$ vyjadruje priemernú hodnotu druhej mocniny gradientov v čase t .

RMSprop, je adaptívne učiaci sa metóda vytvorená Geoffom Hintonom. Jedná sa o metódu takmer totožnú s Adadelta metódou. Snahou obidvoch metód je riešiť nedostatky metódy Adagrad. V publikovanom článku od G. Hintona sú odporúčané hodnoty 0,9 pre γ a 0,001 pre α . [42]

V tejto kapitole bolo spomenutých 8 optimalizačných algoritmov. Všetky z nich majú spoločný základ rovnice, len sa ku každému navyše pridal člen, ktorý umožnil nové funkcie pre daný algoritmus. Algoritmus RMSprop má odstránené mnohé nedostatky v porovnaní ostatných optimalizačných algoritmov. Napriek mnohým vylepšeniam sa nedá povedať, že je vhodný na riešenie každého problému. Všetky vyššie uvedené algoritmy majú svoje využitie a tým pádom sa nedá jasne povedať, ktorý je najlepší. Hľadanie vhodnej metódy pre konkrétnu úlohu je viac empirický, než matematický postup.

2.2.9 Backpropagation (Spätná propagácia)

Spätná propagácia umožňuje počítať gradient chybovej funkcie v závislosti na váhach neurónovej siete. Prostredníctvom metódy spätnej propagácie sa počíta negatívny gradient chybovej funkcie vo vzťahu k váham neurónovej siete [43]. V priebehu učenia neurónovej siete sa iteratívne upravujú váhy a tým sa neurónová sieť učí. Na získanie informácie, ako veľmi je treba upraviť jednotlivé váhy, sa používa výstupná chyba v kombinácii s negatívnym gradientom chybovej funkcie. V jednoduchnej sieti obsahujúcej jeden neurón je možné určiť priame spojenie váh s chybou na výstupe z neurónovej siete pomerne jednoducho. Pri zložitých neurónových sieťach, ktoré obsahujú niekoľko vrstiev s mnohými neurónmi, je určenie závislosti váh na výsledku výpočtovo náročnejšie.



Obr. 6: Diagram spätnej propagácie

Proces spätnej propagácie spočíva najprv v doprednej propagácii, kde sa získajú hodnoty chýb. Hneď po doprednej propagácii nasleduje spätná propagácia. Algoritmus spätnej propagácie by sa mohol rozdeliť na dva celky. V prvej časti sa počítajú váhy, ktoré vstupujú do výstupných neurónov, nakoľko výstupné neuróny sú závislé len na jednej chybovej funkcii. Neuróny, ktoré sa nachádzajú v skrytej vrstve neurónovej siete, sú závislé na všetkých výstupných chybových funkciách. Neuróny v hlbších vrstvách sa zaradia do druhej skupiny v algoritme spätnej propagácie.

V prvej skupine sa počas doprednej propagácie získajú hodnoty chybových funkcií. Následne sa hľadá závislosť váhy vstupujúcej do neurónu, s požadovaným výstupom z neurónovej siete. Na získanie závislosti sa používa reťazové pravidlo, kde sa postupne derivujú jednotlivé zložky od výstupnej chyby až k váham neurónovej siete. Skladá sa z troch derivácií. Najprv sa derivuje chybová funkcia podľa výstupu z daného neurónu. Druhou

deriváciou v poradí je výstup z neurónu podľa sumácie všetkých váh a biasov. Nakoniec sa derivuje sumácia podľa príslušnej váhy vstupujúcej do neurónu. Tento proces sa vykoná pre všetky výstupné neuróny. Rovnica výpočtu parciálnej derivácie chybovej funkcie vzhľadom na príslušnú váhu neurónu:

$$\frac{\partial C_{cel}}{\partial w_5} = \frac{\partial C_{cel}}{\partial y_{y1}} \frac{\partial y_{y1}}{\partial net_{y1}} \frac{\partial net_{y1}}{\partial w_5} \quad (35)$$

kde w_5 je váha vstupujúca do výstupného neurónu, y_{y1} je výstup z poslednej vrstvy, net_{y1} je suma všetkých vstupujúcich váh do výstupnej aktivačnej funkcie a C_{cel} je celková chyba neurónovej siete na výstupe. [44]

Druhá skupina má podobný proces. Rozdiel je v počte výstupov chybových funkcií. Je potrebné zahrnúť všetky chybové funkcie zo všetkých výstupných neurónov. Vzťah váhy na celkovú chybu sa počíta reťazovým pravidlom, podobne ako to bolo v prvej skupine. Výstup z vnútorného neurónu má vplyv na celkové chyby z výstupných neurónov. Prvý člen v rovnici (27) reprezentuje súčet všetkých výstupných derivácií chybovej funkcie podľa výstupu:

$$\frac{\partial C_{cel}}{\partial w_1} = \frac{\partial C_{cel}}{\partial y_{s1}} \frac{\partial y_{s1}}{\partial net_{s1}} \frac{\partial net_{s1}}{\partial w_1} \quad (36)$$

kde w_1 je váha vstupujúca do neurónu v skrytej vrstve, y_{s1} je výstup zo skrytej vrstvy, net_{s1} je súčet všetkých vstupujúcich váh do aktivačnej funkcie v neuróne zo skrytej vrstvy a C_{cel} reprezentuje celkovú chybu na výstupe z neurónovej siete. Rovnica (37) vyjadruje súčet jednotlivých derivácií v skrytej vrstve neurónovej siete. Po úpravách sa rovnica (36) môže vyjadriť aj vo forme rovnice (38).

$$\frac{\partial C_{cel}}{\partial y_{s1}} = \sum \frac{\partial C_{yi}}{\partial y_{s1}} \quad (37)$$

$$\frac{\partial C_{cel}}{\partial w_1} = \frac{\partial y_{s1}}{\partial net_{s1}} \frac{\partial net_{s1}}{\partial w_1} \left(\sum \frac{\partial C_{cel}}{\partial y_y} \frac{\partial y_y}{\partial net_y} \frac{\partial net_y}{\partial w_1} \right) \quad (38)$$

Finálnym krokom pri spätnej propagácii je zmena pôvodných váh na hodnoty získané počas spätnej propagácie. Dôležité je vykonať zmenu hodnôt váh až po spätnej propagácii [45]. Hodnota váhy sa upravuje podľa rovnice:

$$w_{t+1} = w_t - \alpha \frac{\partial C_{cel}}{\partial w} \quad (39)$$

kde w_{t+1} je upravená váha neurónovej siete, w_t je predchádzajúca váha neurónovej siete, α je koeficient učenia a $\frac{\partial C_{cel}}{\partial w}$ vyjadruje parciálnu deriváciu celkovej výstupnej chyby vzhľadom k danej váhe neurónovej siete. [44]

2.3 Príprava dát na tréovanie neurónovej siete

Dôležitou súčasťou strojového učenia je použiť správne dáta. Neurónová sieť sa učí na základe dát, ktoré do sústavy vložíme. Pred tréovaním neurónovej siete je potrebné upraviť dáta na formu vyhovujúcu danej neurónovej sieti. V procese prípravy dát na tréovanie neurónovej siete dochádza najmä k nižšie uvedeným problémom:

- nekompletnosť dát
- nečistota dát (šum)
- nekonzistencia dát

V prípade náhodnej nekompletnosti dát, je jednou z možností úplné odstránenie daného riadku zo zoznamu veličín. Nevýhodou tohoto spôsobu je potenciálna strata množstva dát. Ďalším problémom by mohla byť nekompletnosť dát spôsobená neznámym systémom.

Hlboké neurónové siete sú citlivé na vstupné hodnoty. V záujme zlepšenia výkonu neurónovej siete sa vstupné dáta normalizujú na predom určenú hodnotu. Využívajú sa dva spôsoby: [46]

- **Normalizácia** je úprava rozsahu hodnôt pôvodných dát na hodnoty od 0 do 1. Najčastejšie používaným spôsobom je min-max. Dáta sa zoradia od najmenšieho, reprezentované číslom 0, až po najväčšie, reprezentované číslom 1.
- **Štandardizácia** spočíva vo výpočte aritmetického priemeru z používaných dát. Jednotlivé hodnoty sa upravujú tak, aby sa vyskytovali okolo priemeru, s maximálnou deviáciou 1.

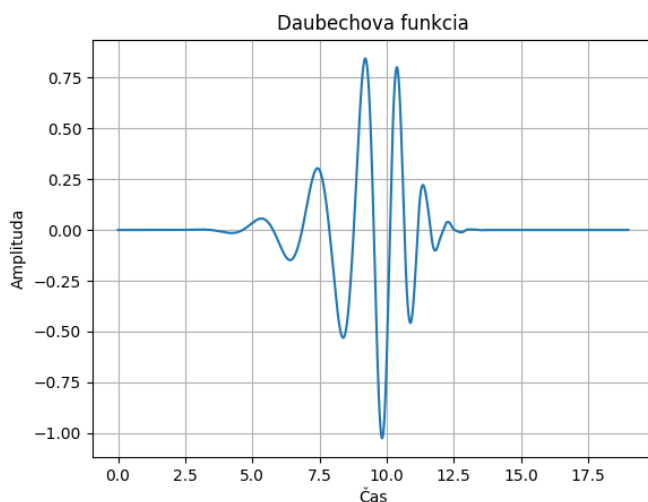
Ďalšou z vhodných príprav dát na tréovanie neurónovej siete je extrakcia atribútov (*feature extraction*). Extrakcia atribútov slúži na zredukovanie vstupného množstva dát a na výber vhodne reprezentujúcich atribútov. V prípade veľkého datasetu prichádza do úvahy redukcia vstupných dát do neurónovej siete. Proces extrakcie atribútov z analyzovaného systému sprostredkuje dáta, lepšie reprezentujúce skúmané vlastnosti z pôvodných dát. Existuje niekoľko metód na extrakciu vhodných atribútov. Nakoľko táto práca sa zaoberá 1D signálom, budú spomenuté nižšie uvádzané metódy.

V časovej oblasti sa dáta na tréovanie neurónovej siete získavajú priamo z dát získaných prostredníctvom meracích prístrojov. Meracie prístroje poskytujú časový záznam z vibrácií alebo iných senzorov. V tejto práci sa používal časový záznam z akcelerometra, dynamometra a akustického senzora. Po získaní dát z časovej oblasti sa môžu použiť priamo na tréovanie neurónovej siete alebo sa môžu ďalej upravovať. Kapitola 3.3 tejto práce sa zaoberá priamo s neupravenými dátami, získanými z meracích prístrojov. V kapitole 3.5 sa časový záznam upravuje, resp. extrahujú sa štatistické hodnoty z časového záznamu meraných veličín.

Vo frekvenčnej oblasti je záznam z časovej oblasti prevedený do frekvenčnej oblasti prostredníctvom Furiérovej transformácie. Veľká rada vibračných analýz sa vykonáva práve prostredníctvom frekvenčnej analýzy [47]. Transformácia záznamu z časovej do frekvenčnej oblasti, v porovnaní použitia záznamu priamo v časovej oblasti, je výpočtovo náročnejšia.

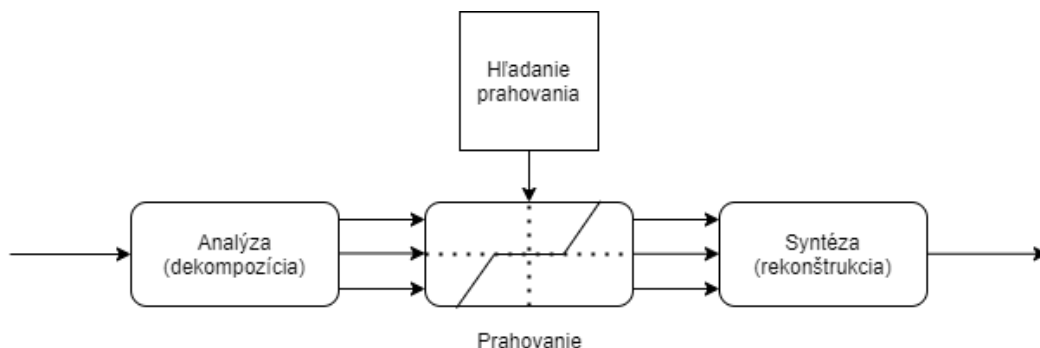
2.4 Filtrácia dát pomocou Wavelet transformácie

Pod pojmom filtrácia v oblasti signálového spracovania, rozumieme odstraňovanie nežiadúcich zložiek zo signálu. Počas filtrácie je snahou odstrániť, resp. zredukovať nechcené elementy v spracovávanom signáli. V oblasti spracovania signálov existuje niekoľko metód, ktoré sa používajú na filtráciu. Jednou z takýchto metód je wavelet. Transformácia typu wavelet je lokalizovaná v čase a frekvencii. Dalo by sa povedať, že sa jedná o zlepšený spektrogram. Wavelet transformácia je vysoko lokalizovaná v čase a frekvencií. Tým pádom je schopná reprezentovať náhle zmeny v analyzovanom signáli. Wavelet je vlna, ktorej dĺžka je ohraničená a priemerná hodnota oscilácie je nulová. [48]



Obr. 7: Vykreslenie Daubechovej wavelet funkcie

Na odstránenie šumu zo signálu pomocou wavelet transformácie sa používa algoritmus znázornený na obr. 8. V prvej časti sa zašumený signál rozloží na jednotlivé koeficienty. V ďalšom kroku sa signál prahuje. Level prahovania sa hľadá prostredníctvom metód, ako je VishuShrink, Bayes, SURE. Samotné prahovanie je založené na metóde *Hard* alebo *Soft*. V prípade *Hard* metódy sú vlnové koeficienty v rozmedzí $-\lambda$ a $+\lambda$ nulové. Hodnoty mimo tento rozsah sú zachované v pôvodnom stave. Metóda *Soft* je rozdielna v posunutí hodnôt λ . [49]



Obr. 8: Schéma procesu filtrácie pre wavelet funkcie v balíčku pywt

2.5 RUL

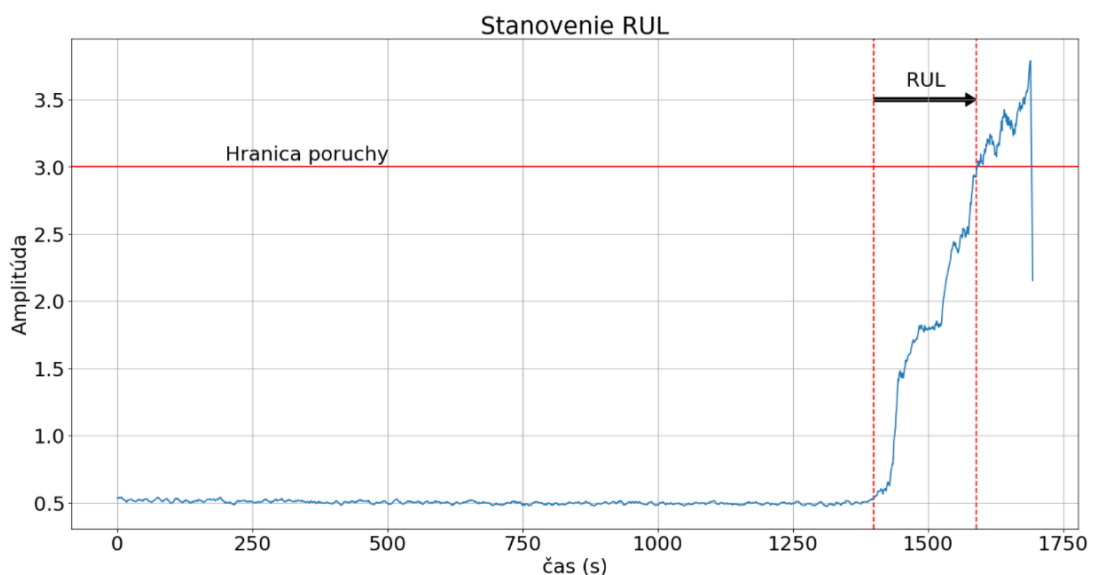
RUL (*Remaining useful life*) reprezentuje dobu životnosti monitorovanej súčasti. Snahou RUL je včasné zachytenie opotrebenia a následné naplánovanie výmeny opotrebenej, resp. poškodenej časti. Použitie metódy RUL je predikčnou formou údržby strojov. Obsluhou stroja je možné naplánovať odstávku, na základe stavu opotrebenia jednotlivých komponentnou monitorovaného stroja. Metódy RUL sa delia na základe dostupnosti dát na: [50]

Run-to-Failure metóda je určená na predikciu ostávajúceho života monitorovaných súčastí na základe vopred známej databázy predošlých RUL. Spomenutá metóda vyžaduje vopred známu databázu, ktorá obsahuje záznamy o priebehu opotrebenia monitorovanej časti. Priebehy z databázy sú porovnávané s predikciou neurónovej siete. Následne sa hľadá priebeh opotrebenia monitorovanej časti, ktorý je najpodobnejší získanej predikcii z neurónovej siete. Vybraný priebeh života z danej databázy je ďalej použitý na predikciu RUL.

Threshold metóda určovania RUL spočíva v stanovení ohraničujúcej hodnoty, ktorá reprezentuje dosiahnutie konca funkčnej doby monitorovanej súčasti. Hodnota RUL sa stanoví ako rozdiel času, v ktorom nastane porucha a doby začatia poruchy. Rovnica pre výpočet RUL:

$$RUL = t_f - t_s \quad (40)$$

kde t_f je čas, kedy nastane porucha a t_s je moment, v ktorom je predikované RUL. Neurónová sieť pri detekcii počiatočného opotrebenia začne predikovať zostávajúcu dobu životnosti časti až do dosiahnutia okrajovej podmienky. Okrajová podmienka v tomto prípade reprezentuje dosiahnutie konca životnosti monitorovanej súčasti. Metóda je vhodná pre časti, ktorých opotrebenie má podobný priebeh. Taktiež na použitie tejto metódy nie je potrebné rozsiahle množstvo predošlých meraní priebehu životnosti monitorovaných častí.



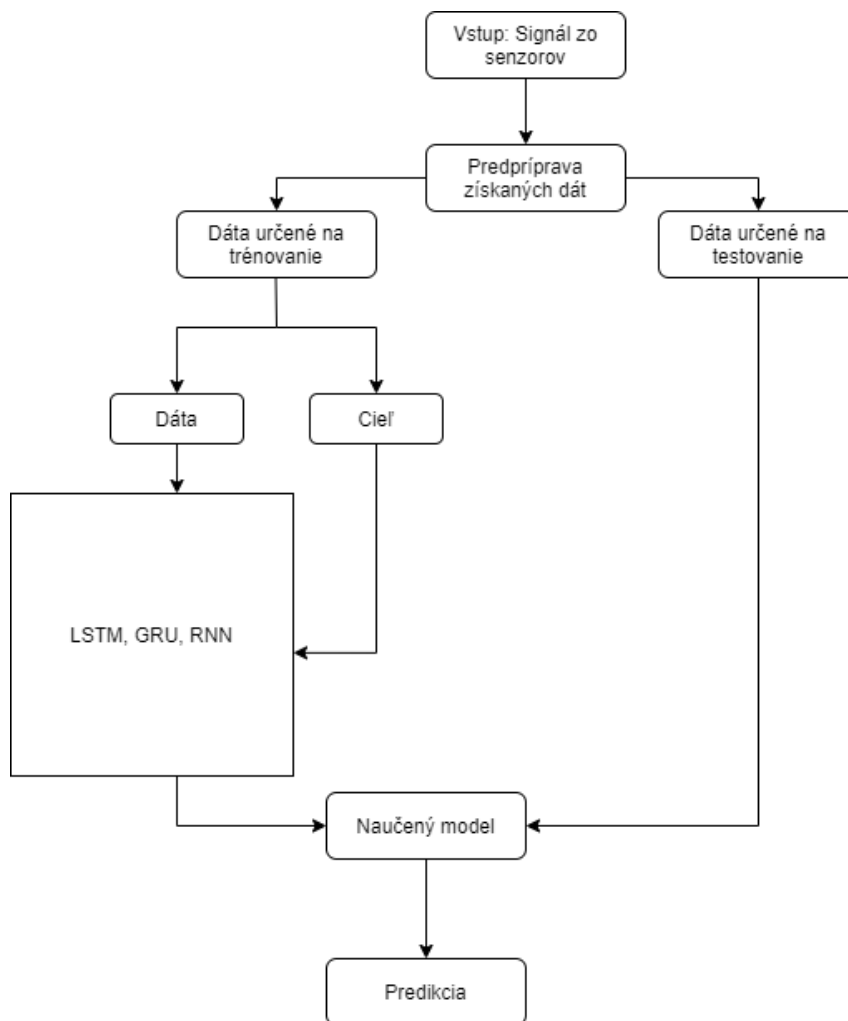
Obr. 9: Znáznornenie Threshold metódy na určenie RUL

3 Praktická časť monitorovania obrábacieho stroja

Cieľom praktickej časti bolo monitorovanie komponentov obrábacieho stroja za použitia rekurentnej neurónovej siete. Dôvodom výberu použitia rekurentnej neurónovej siete bol fakt, že sprostredkované dáta mali sekvenčný charakter. Všetky získané dáta boli v časovej oblasti, z verejne dostupných zdrojov. V časovej oblasti dáta na seba nadväzujú a rekurentná neurónová sieť je schopná zachytiť tendenciu monitorovanej sekvencie. Dáta boli získavané zo senzorov dynamometra, akcelerometra a akustického senzora.

3.1 Základný prehľad použitých algoritmov

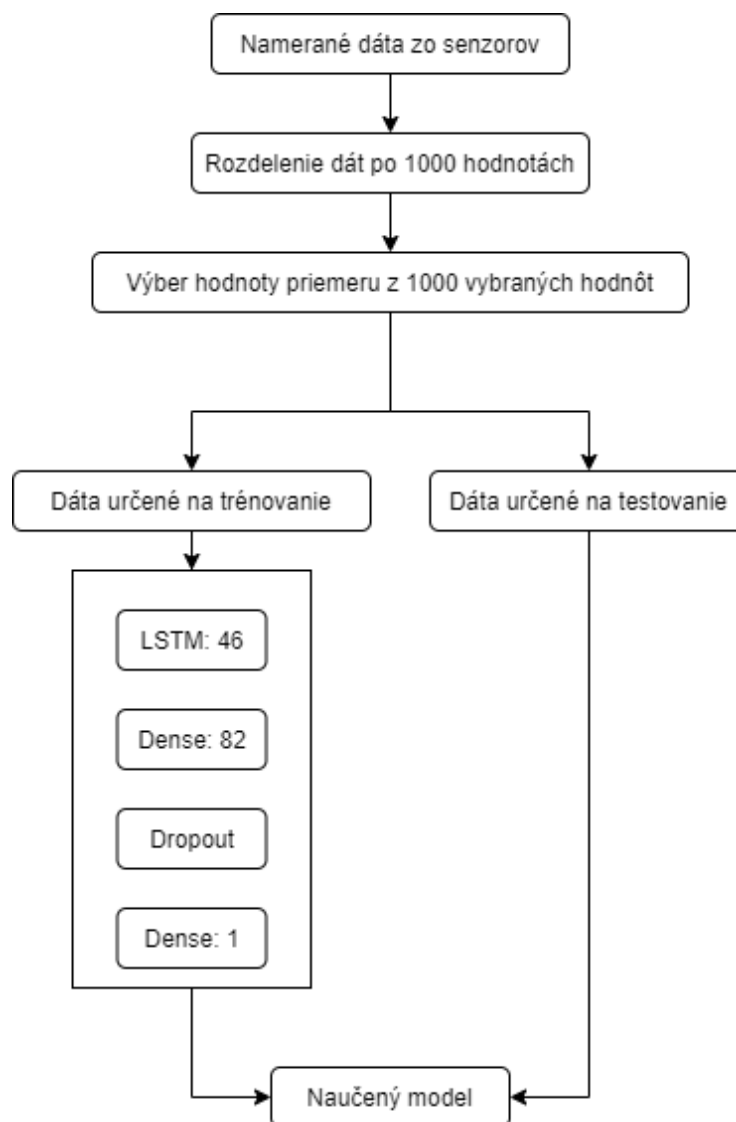
Na trénovanie neurónovej siete bola zvolená metóda s učiteľom. Pôvodné dáta boli rozdelené do dvoch častí. Prvá časť slúžila na trénovanie neurónovej siete. Druhá časť testovala schopnosť predikcie neurónovej siete. V ďalších troch podkapitolách sú vysvetlené použité typy algoritmov určené na monitorovanie stavu komponentov strojov. V každom z algoritmov sa používa rekurentná neurónová sieť a jej modifikácie, akými sú LSTM a GRU.



Obr. 10: Schéma trénovania a testovania neurónovej siete

3.1.1 Algoritmus určovania momentálneho stavu nástroja

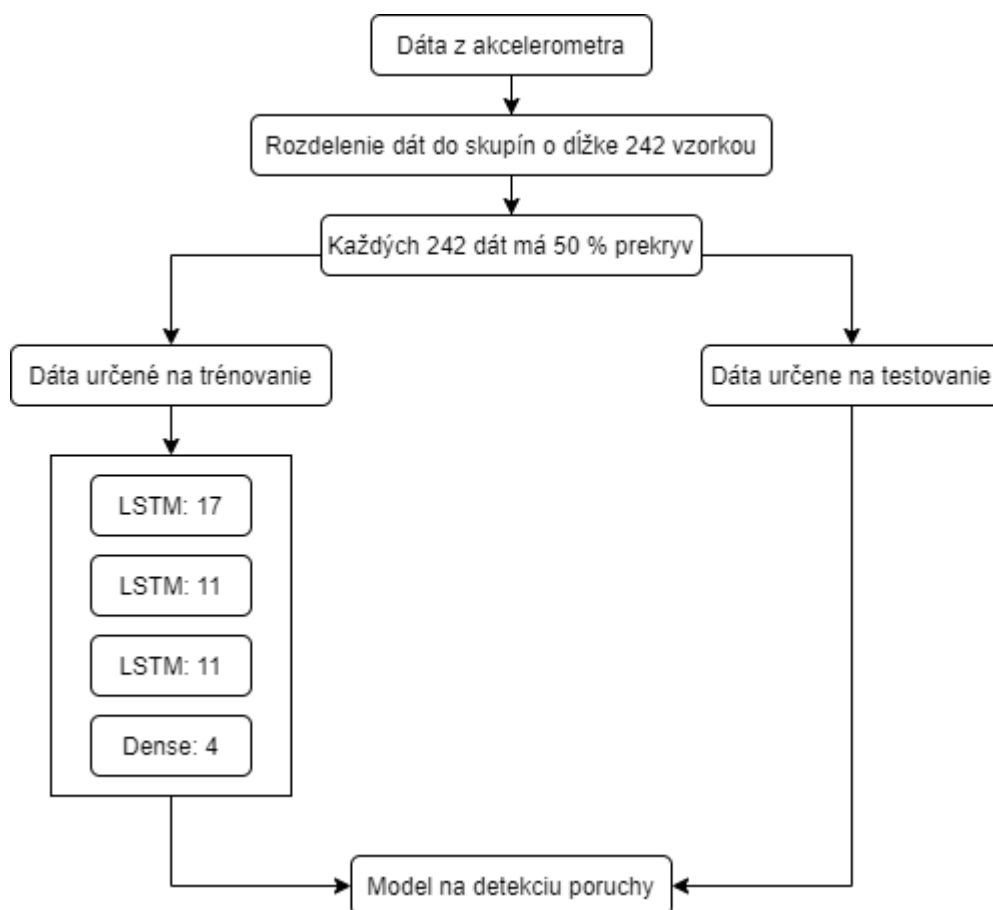
Na obr. 11 je znázornený algoritmus na nepriame určovanie stavu obrábacieho nástroja. Dáta zo senzorov sú používané na stanovenie momentálneho stavu opotrebenia nástroja. Pôvodné dáta boli zredukované pomocou rozdelenia dát do skupín o 1000 prvkoch. Následne sa z každej skupiny vybrala priemerná hodnota. Získané hodnoty priemeru sa následne združili do dát na trénovanie a testovanie použitého algoritmu. Dáta určené na trénovanie neurónovej siete boli vkladané do rekurentného typu neurónovej siete. Samotná architektúra neurónovej siete sa skladala z 3 vrstiev (vid'. Obr. 11).



Obr. 11: Algoritmus na stanovenie stavu frézy

3.1.2 Algoritmus na detekciu a klasifikáciu poruchy

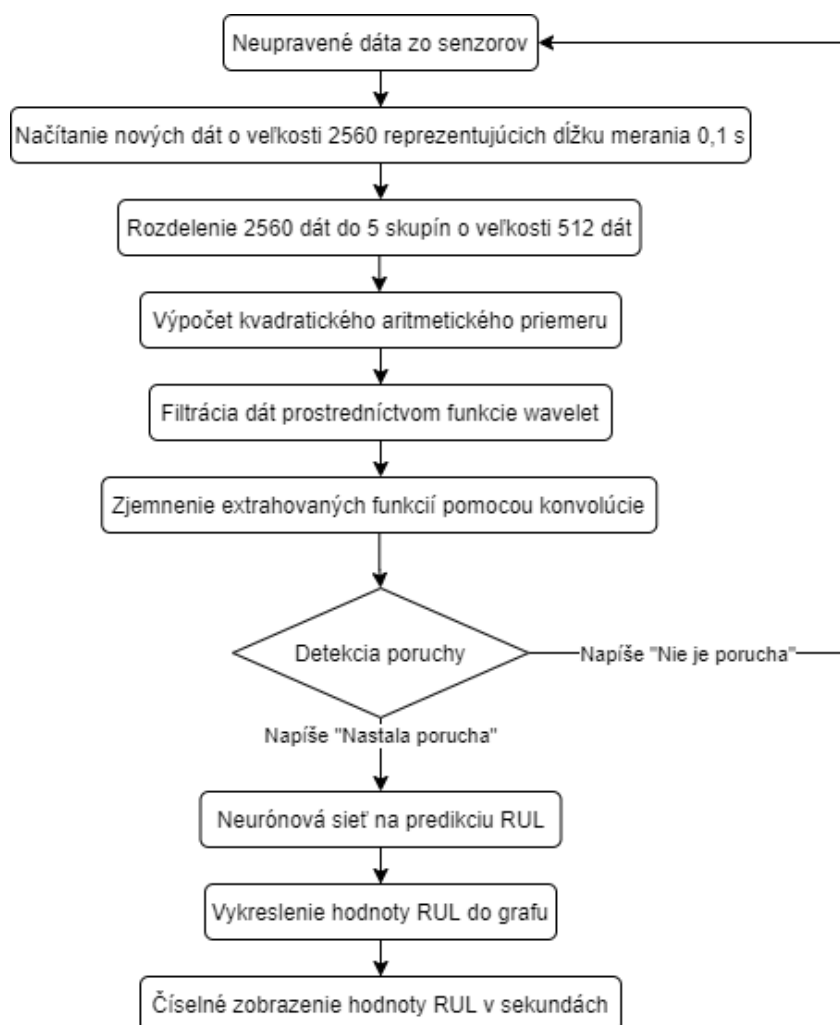
Algoritmus je určený na klasifikáciu nameraných dát do 4 skupín. Prvá skupina reprezentuje dáta z normálneho stavu monitorovania jedného ložiska. Ďalšie 3 skupiny reprezentujú typ poškodenia ložísk. Poruchy ložiska sa klasifikujú na základe časti poškodenia ložiska. Zvolený algoritmus je schopný klasifikovať, či detegovať danú poruchu. Dáta získavané priamo z akcelerometra sú rozdelené do skupín s dĺžkou časového záznamu 242 prvkov. Takto rozdelené skupiny majú polovičné prekrytie dát s nasledujúcou skupinou. Ďalším rozdelením bolo delenie do dvoch skupín. Jedna skupina bola určená na tréning neurónovej siete. Druhá skupina dát bola určená na validáciu natrénovaného modelu neurónovej siete. Zvolená rekurentná neurónová sieť sa skladala zo 4 vrstiev. Prvé 3 vrstvy boli typu LSTM rekurentnej neurónovej siete. Výstupná vrstva bola typu *dense*, resp. plne prepojenej siete o veľkosti štyroch neurónov. Použitá aktivačná funkcia na výstupnej vrstve bola zvolená softmax, určená na klasifikáciu výstupu z neurónovej siete.



Obr. 12: Klasifikácia stavu monitorovaného ložiska

3.1.3 Algoritmus na monitorovanie a predikciu stavu ložiska

Použitý algoritmus pozostáva z dvoch častí. Prvá časť bola určená na detekciu poruchy monitorovaného ložiska. Program zo získaných dát rozhodoval o stave ložiska. V prípade detekcie poruchy ložiska sa spustila druhá časť programu. Úlohou druhej časti programu bolo predikovať zostávajúci čas ložiska do dosiahnutia vopred stanovenej hodnoty zrýchlenia získavaného z akcelerometrov. Táto hodnota vyjadrovala predikciu stavu ložiska, ktoré je určené na výmenu. Dáta boli získané zo súboru typu CSV. V prípade online monitorovania by sa namerané dáta vpisovali do CSV súboru priamo z akcelerometra. Program prečítal z akcelerometra vždy 2560 dát. Tieto dáta reprezentovali dobu jedného merania algoritmu o dĺžke 0,1 s. Pričom dáta sa periodicky načítavali po uplynutí 10 s. Následne algoritmus rozdelil dáta do skupín o dĺžke 512 hodnôt. Z takto získaných skupín sa počítala hodnota kvadratického aritmetického priemeru. Ďalším krokom bolo vyhodnotenie stavu ložiska. Na to bola použitá neurónová sieť, ktorá mala za úlohu detegovať začiatok opotrebenia ložiska. V prípade detekcie začiatku opotrebenia sa spustila druhá časť programu, ktorá predikovala zostávajúcu dobu do konečného poškodenia.



Obr. 13: Schéma algoritmu na monitorovanie a predikciu stavu monitorovaného ložiska

3.2 Popis riešených problémov v praktickej časti práce

V tejto práci bolo vytvorené riešenie troch problémov na monitorovanie stavu obrábacieho stroja. Snahou je zabezpečiť v priemysle komplexnejšiu kontrolu strojov. Nakoľko sa stroje skladajú z niekoľkých častí, nie je postačujúce kontrolovať len ich jednu časť. Cieľom je predchádzať poruchám. V prípade vzniku poškodenia by bola nutná odstávka, čo by malo za následok veľké finančné straty. Stroj mimo prevádzky brzdi celý proces výroby. V súčasnej dobe na riešenie práve takýchto problémov je snaha využívať technológiu IoT. V ďalších kapitolách na monitorovanie stavu stroja sú riešené tri úlohy využívajúce umelú inteligenciu. Informácie ohľadom predikcie stavu boli získavané zo senzorov v podobe sekvenčných dát. Problém číslo 1 tejto práce sa venuje fréze, konkrétne sleduje stav obrábacieho nástroja frézky. V probléme 2 a 3 sa rozoberá monitorovanie ložísk. Problém 2 má za úlohu detegovať a identifikovať poruchu na ložisku. Problém 3 sa venuje predikcii zostávajúcej životnosti ložiska (RUL).

3.3 Problém 1 - Predikcia stavu frézy

Dôležitou súčasťou obrábacieho stroja je práve obrábací nástroj. Nástroj, ako je napríklad fréza, sa dostáva do priameho kontaktu s iným telesom, čo spôsobuje jeho opotrebenie. V priemysle až 20 % odstávok stroja je spôsobených nadmerným opotrebením obrábacích nástrojov [51]. Presné monitorovanie opotrebenia časti stroja by potenciálne mohlo zredukovať náklady až o 30 % [51].

Na monitorovanie stavu nástroja sa používajú dve metódy, priama a nepriama. **Priama metóda** využíva mikroskop, ktorým sa priamo zisťuje opotrebenie nástroja [52]. V praxi je zložité realizovať priamu metódu. Obsluha musí manuálne vybrať nástroj a získať informáciu o stave komponentu. Odstávka spôsobuje finančné straty, nakoľko stroj nemôže vykonávať svoju činnosť. **Nepriama metóda** spočíva v monitorovaní stavu frézy prostredníctvom senzorov, ako je akcelerometer alebo dynamometer. Pri nepriamej metóde sa priamo nemonitoruje opotrebenie komponentu. Hodnoty zo senzorov akcelerometru a dynamometru sa využívajú na aproximáciu stavu komponentu.

3.3.1 Rozpis datasetu opotrebenia CNC stroja

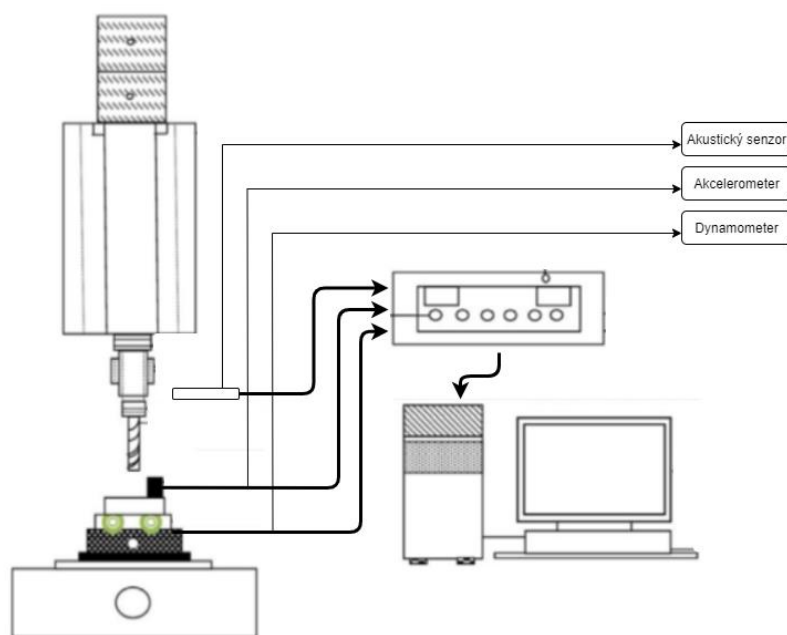
Dáta boli získavané z CNC frézky. Na obrobok boli umiestnené tri akcelerometre, merajúce vibrácie nástroja v x -ej, y -ej a z -ej osi. Ďalším zo senzorov bol dynamometer, ktorý bol umiestnený medzi obrobok a stôl CNC stroja. Tretím typom senzoru bol akustický senzor AE, merajúci hluk, spôsobený opotrebením nástroja. Experiment spočíval v procese obrábania konštantnej plochy obrobku. Počas procesu sa zo senzorov zaznamenávali dáta. Po frézovaní požadovanej plochy obrobku bola vybratá fréza a pomocou mikroskopu LEICA MZ12[53] bola zistená hĺbka opotrebenia nástroja. Hodnota hĺbky opotrebenia sa priradila k záznamu, ktorý bol získaný počas frézovania z vyššie uvedených troch druhov senzorov. Proces obrábania a následného merania opotrebenia mikroskopom bol opakovaný 315 krát. V datasete sa nachádzajú tri skupiny meraní C1, C4 a C6 [51]. V každom z nich sa nachádza 315 meraní. Bližšie informácie o meraní sú v tab. 2. Na zachytávanie informácií zo senzorov bol použitý DAQ NI PCI1200 [53]. Vzorkovacia frekvencia počas merania opotrebenia stroja bola 50 kHz.

V každom časovom kroku je sedem hodnôt. Jednotlivé hodnoty sú získavané z vyššie spomenutých senzorov, ako je akcelerometer, atď. [53]

Tab. 2: Hodnoty základných parametrov obrábacieho stroja [53]

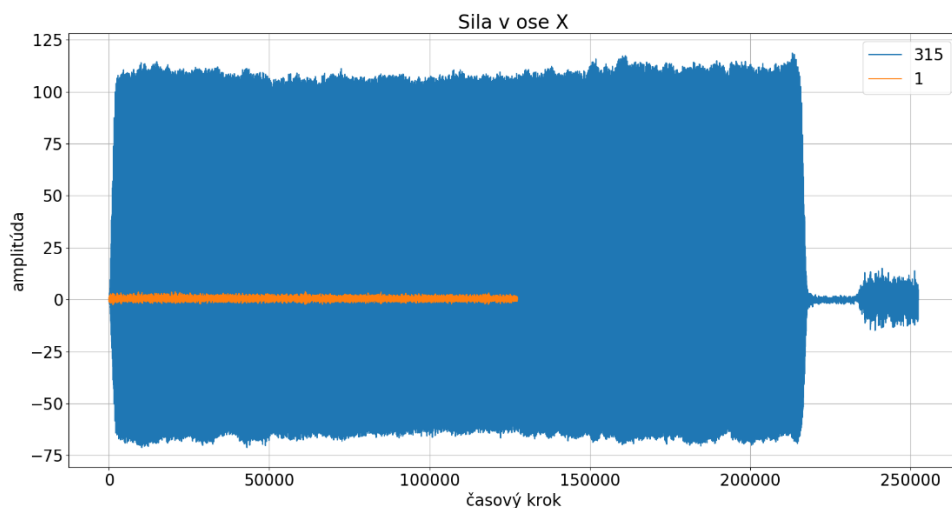
Parametre obrábacieho procesu	Hodnoty
Rýchlosť posuvu	1555 mm/min
Hĺbka rezu	0,125 mm
Hĺbka rezu	0,2 mm
Rýchlosť otáčania vretena	10400 ot/min
Obrábací nástroj	fréza s guľovým nosom z karbidu volfrámu

Na obr. 14 je nákres CNC frézy a schéma experimentu. Zobrazené sú tri akcelerometre, ktoré merajú vibrácie v troch osiach. Medzi obrobkom a stolom CNC stroja sa nachádza dynamometer. Hodnoty z dynamometra sú trojdimenzionálne z *x*-ej, *y*-ej a *z*-ej osi. Nakoniec pomocou akustického senzoru sa získavajú hodnoty hluku.

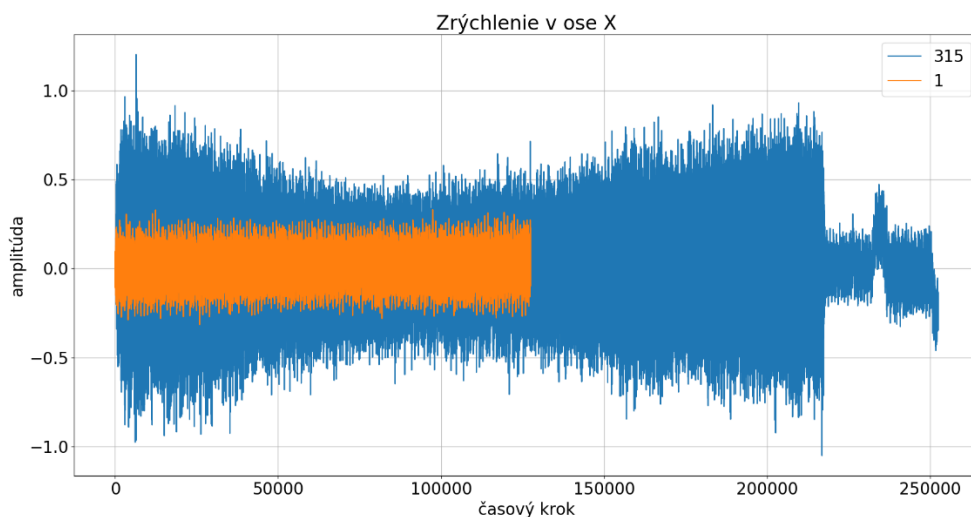


Obr. 14: Schéma merania dát na predikciu opotrebenia frézy [54]

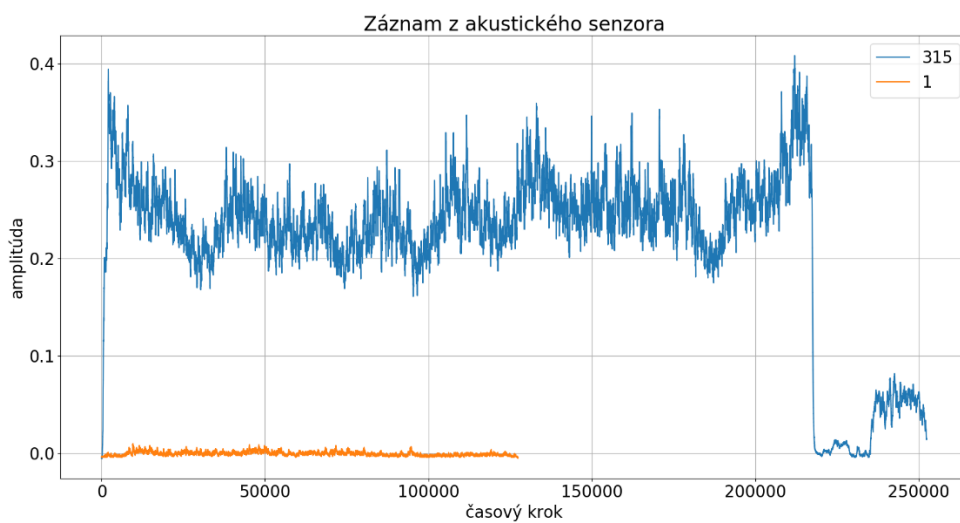
Poskytnuté dáta z merania boli v tejto práci použité na učenie neurónovej siete. Cieľom bolo získať vhodnú metódu na monitorovanie stavu obrábacieho nástroja. Správna identifikácia by zefektívnila celý proces výroby. Spôsob merania dát je neinvazívny. Vďaka ich neinvazívnej povahe sú vhodné na monitorovanie stavu stroja počas prevádzky. Ako je vidieť na obr. 15, je výrazný rozdiel v amplitúdach signálového záznamu z prvého a tristo pätnásteho merania. Najvýraznejší rozdiel je medzi hodnotami získanými z dynamometra a akustického senzora. Naopak rozdiel amplitúd získaných z akustického senzora je v porovnaní dynamometra najmenší.



Obr. 15: Porovnanie signálu z dynamometra pre 1. meranie a pre 315. meranie

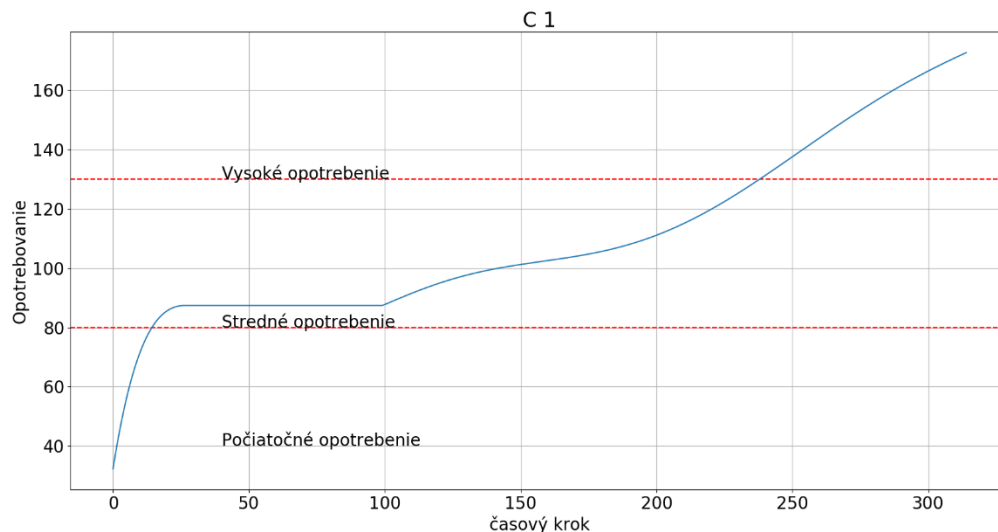


Obr. 16: Porovnanie signálu z akcelerometra pre 1. meranie a 315. meranie

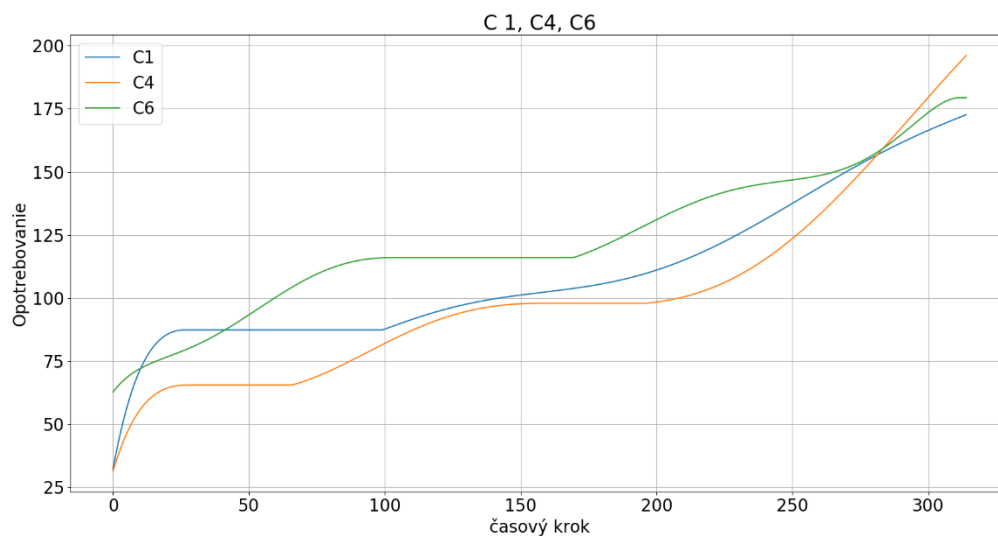


Obr. 17: Porovnanie signálu z akustického senzora pre 1. a 315. meranie

Najvýraznejšie opotrebenie je v zázname z x -ej osi dynamometra. V každom zázname sa postupne zvyšujú amplitúdy senzorových signálov. Najčastejším opotrebením pre makro nástroje je abrazívne opotrebenie [55]. V prípade, ak sa dve telesá o seba trú, vzniká abrazívne opotrebenie. Typická krivka opotrebenia pre obrábacie nástroje sa skladá z troch častí. V prvej časti dochádza k prudkému počiatocnému opotrebeniu. Po prudkom náraste opotrebenia nasleduje ustálený stav (druhá časť), pri ktorom dochádza k pomalému opotrebeniu nástroja. Najdlhší čas obrábania stroja sa odohráva počas ustáleného stavu. Kritické opotrebenie nastáva na konci ustáleného opotrebenia. Jedná sa o prudký nárast poškodenia obrábacieho nástroja. V tomto štádiu dochádza k poškodeniu frézy [56]. V prípade obrábacích nástrojov nie je vhodné používať ich v tejto oblasti. Nástroje nie sú schopné dosahovať takých presností obrábania, ako v predchádzajúcich dvoch stavoch [56]. Na obr. 19 je vidieť podobnosť troch meraní C1, C4 a C6. V každom zo záznamov priebehu opotrebovania je možné jednoznačne pozorovať tri sekcie opotrebenia. Priebehy C1, C4 a C6 sa líšia rozdielnou dobou trvania jednotlivých sekcií, akými sú počiatocné, stredné (ustálené) a vysoké (finálne) opotrebenia.



Obr. 18: Graf opotrebenia frézy rozdelený do troch sekcií



Obr. 19: Porovnanie troch meraní opotrebenia

3.3.2 Postup riešenia monitorovania stavu obrábacieho nástroja

Dáta zo senzorov boli použité na trénovanie neurónovej siete so základnou architektúrou rekurentnej siete. V každom časovom kroku boli vstupné dáta 7- prvkové. Každá zo siedmych prvkov reprezentovala hodnotu z jedného zo senzorov, a to akustický senzor, akcelerometer a dynamometer. Dáta majú charakter časovej sekvencie. V prípade rekurentnej varianty algoritmu neurónovej siete nie je nutné ďalej upravovať získané dáta. Je možné zo senzorov použiť priamo získaný záznam. Vďaka tomuto postupu sa urýchlí samotný proces prípravy a následného trénovanie siete. Taktiež je jednoduchšie doučiť sieť, ak sa získajú dodatočné dáta na trénovanie. V prípade priemyselného použitia, pri doučení, je umožnená aj výrazná redukcia času. V našom prípade sa upravil jedine počet hodnôt pre jedno meranie opotrebenia vzorky. Záznam zo senzorov pre takéto jedno meranie obsahoval až 200 000 hodnôt. Na zrýchlenie procesu učenia sa pôvodný záznamu zredukoval. Záznam sa podelil na niekoľko menších zložiek, z ktorých sa získala priemerná hodnota. Zredukovaný záznam bol použitý na učenie siete, využívajúc tri typy algoritmov, a to RNN, LSTM a GRU. Pre spomenuté typy algoritmov bola na výstupe použitá vrstva *dense* s 82 neurónmi a aktivačnou funkciou hyperbolického tangensu (*tanh*). Vrstva *dense* je klasická, hlboko prepojená neurónová vrstva siete. Nasledovala funkcia *dropout* s nastavením na 0,6. Dôvodom použitia funkcie *dropout* bolo zamedzenie pretrénovania siete. Nedostatok dát na trénovanie zväčšil nároky na použitie danej funkcie. Na výstupe bola vrstva hlboko prepojenej neurónovej siete *dense* s jedným neurónom a lineárnou aktivačnou funkciou. Na optimalizáciu pri výpočte spätnej propagácie bol zvolený algoritmus *RMSprop* (*Root mean square prop*). Na získavanie chyby počas porovnávania požadovaného výstupu a aktuálneho výstupu zo siete pri učení, bola zvolená stredná kvadratická chyba. Schéma algoritmu je znázornená v kapitole 3.1.1. Jedna časť dát sa vyhradí na učenie a druhá časť dát sa určí na testovanie.

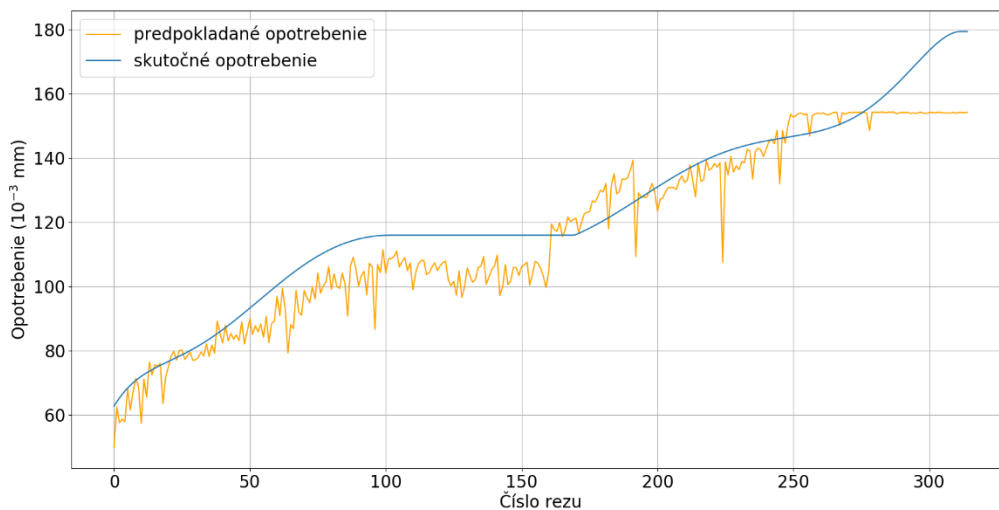
3.3.3 Výsledky neurónovej siete po tréovaní

Na ohodnotenie jednotlivých algoritmov bola použitá hodnota strednej kvadratickej chyby (vid'. Tab. 3). Z porovnaných hodnôt strednej kvadratickej chyby vychádza algoritmus GRU najpresnejšie. Naopak algoritmus využívajúci architektúru RNN má najhoršie výsledky. Algoritmus postavený na architektúre LSTM má vyhovujúce výsledky, nakoľko vie pomerne dobre kopírovať krivku opotrebenia (vid'. Obr. 21). U všetkých troch algoritmov je možné pozorovať neschopnosť predikcie hodnôt opotrebenia nad určitú úroveň. Neurónová sieť využívajúca GRU má toto rozhranie najmenšie. Oblasť, v ktorej sa vyskytuje problém s predikciou, je oblasť zvýšeného poškodenia. Daná oblasť indikuje potrebu výmeny obrábacieho nástroja. Z dôvodu poškodenia nástroja nie je problematické použiť neurónovú sieť na monitorovanie stavu obrábacieho stroja.

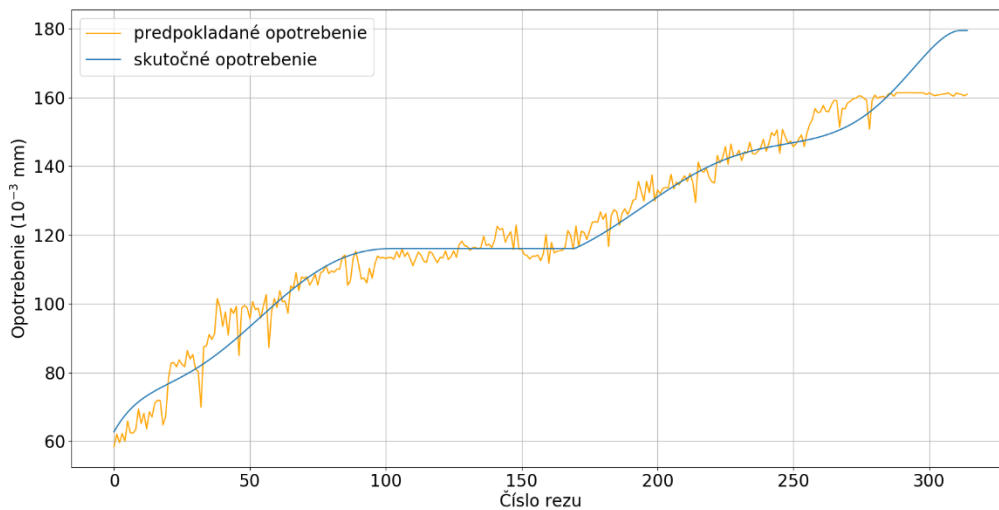
Tab. 3: Výsledky MSE pre použité neurónové siete

Typ siete	Hodnota MSE
RNN	103
LSTM	33
GRU	15

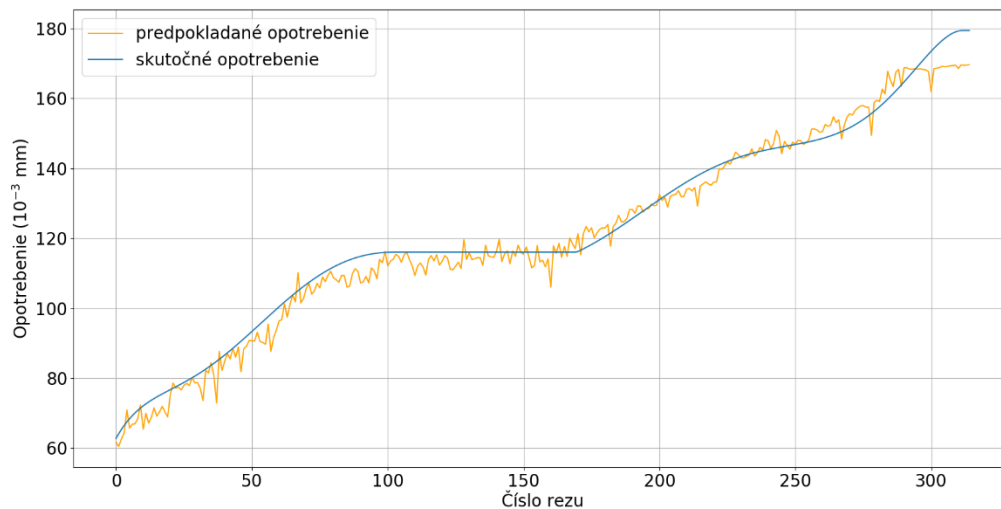
Každá z troch použitých metód je schopná kopírovať krivku vyjadrujúcu opotrebenie nástroja. Treba dodať, že predikované hodnoty z neurónovej siete oscilujú okolo skutočného stavu opotrebenia. Natrénovaný model by sa ďalej dal použiť na stanovenie štádia opotrebenia nástroja alebo na predikciu RUL.



Obr. 20: Predikcia stavu frézy prostredníctvom RNN



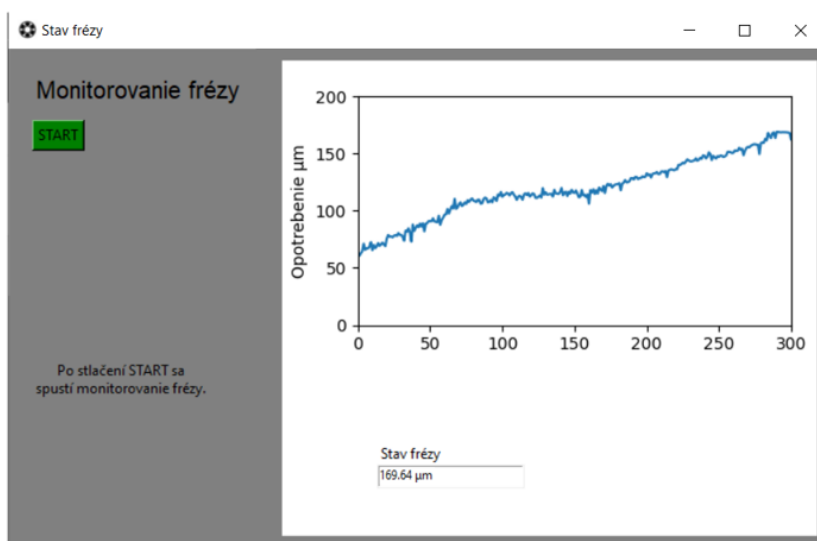
Obr. 21: Predikcia stavu frézy prostredníctvom LSTM



Obr. 22: Predikcia stavu frézy prostredníctvom GRU

3.3.4 Aplikácia na monitorovanie opotrebovania frézy

Na monitorovanie stavu opotrebenia frézy bola vytvorená aplikácia. Program prečíta 128 000 hodnôt zo súboru, v ktorom sú senzorové hodnoty. V prípade reálneho použitia by sa hodnoty vpisovali do súboru, z ktorého by ich program čítal. V ďalšom kroku program počíta priemernú hodnotu z 1000 prvkov. Získaných 128 hodnôt vstupuje do neurónovej siete. Natrénovaná neurónová sieť typu GRU predikuje hodnotu opotrebenia frézy. Hodnoty predikcie sa zobrazia do grafu, a ako číselná hodnota v poli s názvom stav frézy.



Obr. 23: Aplikácia na monitorovanie stavu frézy

3.4 Problém 2 - Detekcia chyby ložiska

Ložiská sú dôležitou súčasťou každého priemyselného stroja. Patria k vysoko namáhaným častiam stroja a preto je dôležité kontrolovať ich stav. Ako bolo spomenuté v kapitole s monitorovaním stavu frézy, aj tu platí, že v prípade odstávky dochádza k finančnej strate. V prípade, ak sa stroj včas nezastaví, hrozí poškodenie obrábacieho stroja. Porušené ložisko môže potenciálne poškodiť ďalšie časti CNC stroja. Cieľom tejto kapitoly je detekcia poruchy ložísk. Natrénovaná neurónová sieť by mala byť schopná rozpoznať poškodené ložisko od nepoškodeného ložiska. Detekcia poškodenia ložiska sa následne môže signalizovať obsluhu daného stroja. Obsluhujúci pracovník môže včas reagovať a vypnúť monitorovaný stroj. Je žiadúce rozpoznať takúto poruchu včas, aby sa konkrétne ložisko mohlo vymeniť za nové.

3.4.1 Dataset porúch ložísk

Hodnoty použité na tréovanie neurónovej siete sú z datasetu poskytnutého Western Reserve University (CWRU) [57]. V tomto súbore sa nachádzajú hodnoty vibrácií získaných z akcelerometrov. Výhodou tohto datasetu na tréovanie neurónovej siete je veľké množstvo dát. Ložiská sú rozdelené podľa typu poškodenia, na poškodenie vonkajšej časti ložiska, na poškodenie vnútornej časti ložiska a na poškodenie valivého elementu. Experiment pozostával z motora o výkone 1,49 kW [57], čím sa roztáčal hriadeľ, na ktorom bolo umiestnené ložisko. Hodnoty z akcelerometrov boli získavané 12 kHz [57] vzorkovacou frekvenciou.

Úlohou ložiska je zabezpečiť prostredie takmer bez trenia a poskytovať oporu a usmernenie pre rotujúce hriadele [58]. Poznáme niekoľko druhov valivých ložísk, ako sú guľové, valčekové, ihlové, súdkové alebo kužeľové ložiská [59]. Ložisko sa skladá z troch častí, a to vnútorného prstenca, vonkajšieho prstenca a vnútornej časti, kde sa nachádza rotujúci element. Z dôvodu vysokého namáhania a priameho kontaktu sa postupne opotrebováva a tým dochádza k poruchám. Každá časť má fyzický kontakt s ďalším telesom. Dochádza k treniu jednotlivých častí a z toho dôvodu môže nastať porucha na akejkolvek časti ložiska. V súbore s dátami získanými z experimentu, sa nachádza niekoľko variant porúch. Jednak sú záznamy z vibrácií podľa poruchy jednotlivých častí ložiska, ale aj poruchy spôsobené pri rozdielnej záťaži na motore.

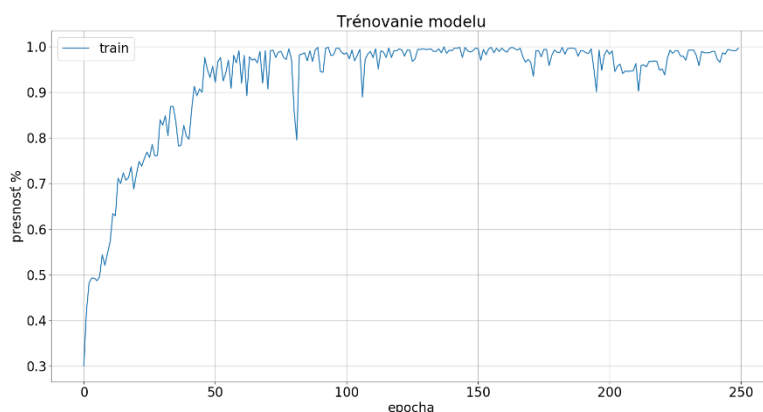
3.4.2 Postup riešenia detekcie poruchy ložiska

Aj v tejto kapitole bola použitá rekurentná neurónová sieť na multiklasifikáciu porúch ložiska. Najpodstatnejšie je rozlíšiť stav ložiska, či je v neporušenom alebo v porušenom stave. V prípade poruchy sa stroj môže hneď zastaviť a ložisko sa vymeniť predtým, ako spôsobí väčšie škody na obrábacom stroji. V tomto prípade sa využila multiklasifikácia, ktorá je schopná identifikovať o akú poruchu ložiska ide. Algoritmus klasifikuje poruchu do štyroch skupín. Pričom prvá skupina je ložisko v normálnom stave. Ďalšie tri skupiny reprezentujú rôzne poruchy ložiska. Na ložisku sa môže poškodiť valivý element alebo vnútorná a vonkajšia časť ložiska. Signál z jednotlivých porúch bol rozdelený na 340 zložiek o dĺžke 242 hodnôt senzoru. Polovica hodnôt z každej zložky mala 50 % prekrytia s nasledujúcou zložkou v poradí. Takto pripravené dáta slúžili ako vstup pre neurónovú sieť. Použitá neuronová sieť bola typu LSTM. Neurónová sieť sa skladala z troch vrstiev. Na výstupe bola použitá funkcia *dense*, ktorá obsahovala 4 neuróny s použitím aktivačnej funkcie *softmax*. Výstupná vrstva s použitím

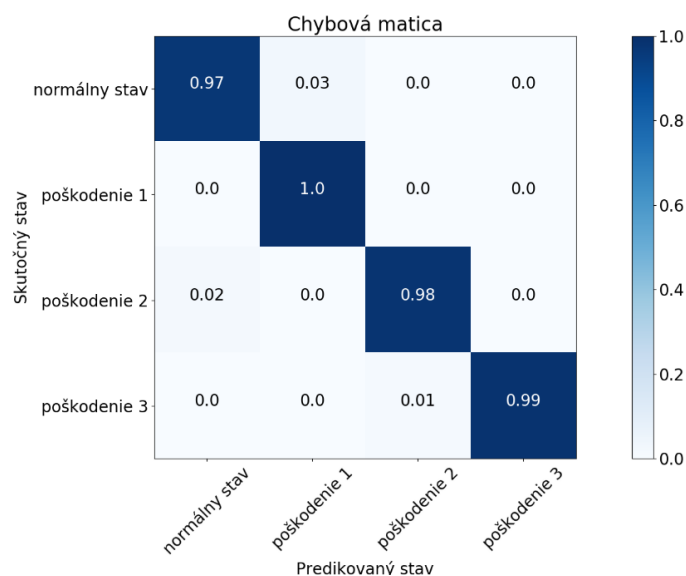
softmax je schopná klasifikovať hodnotu do štyroch kategórií. Ako optimizátor na tréovanie bola použitá funkcia RMSprop. Nakoľko išlo o multiklasifikáciu, bola zvolená kategorická krížová entropická chybová funkcia. Na učenie neurónovej siete bolo zvolených 250 epoch. Po niekoľkých pokusoch bola veľkosť batch stanovená na hodnotu 25.

3.4.3 Výsledky

Na začiatku, pri učení, pomerne výrazne oscilovala presnosť. Približne pri dosiahnutí 200 epoch sa ustálila. Po učení na testovacie dáta bol aplikovaný natrénovaný model. Z dôvodu zamedzenia nesprávnej interpretácie testovacích výsledkov sa dáta rozdelili do dvoch častí. Prvá časť dát bola využitá na tréovanie neurónovej siete. Druhou časťou sa testovala natrénovaná neurónová sieť. Výsledky z testovania sú názorne zobrazené pomocou chybovej matice (vid'. Obr. 25). Veľkosť matice je 4x4. Graficky znázorňuje úspešnosť siete klasifikovať jednotlivé prvky do správnych kategórií. Príslušné triedy klasifikácií sú zoradené v nasledujúcom poradí nepoškodený stav, poškodenie valivého elementu, poškodenie vnútornej a vonkajšej časti ložiska.



Obr. 24: Priebeh tréovania neurónovej siete



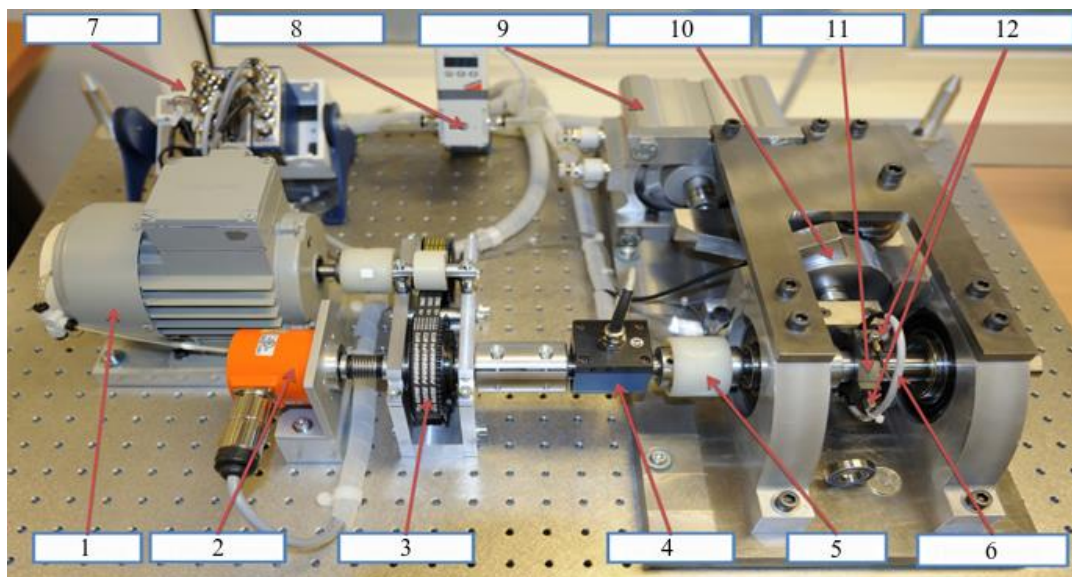
Obr. 25: Chybová matica pre detekciu poruchy ložiska

3.5 Problém 3 - Monitorovanie stavu ložiska

Okrem detekcie poruchy je dôležité predikovať zostávajúci čas životnosti ložiska do vzniku poškodenia. Predchádzajúca kapitola bola zameraná na detekciu poruchy. V tejto kapitole bol navrhnutý algoritmus na predikciu RUL. Na včasnú a preventívnu výmenu ložiska nestačí detegovať poruchu. Pri automatizácii obrábacieho procesu je potrebné mať informáciu o súčasnom stave stroja a aj odhade zostávajúceho života ložiska.

3.5.1 Dataset priebehu života ložiska

Dataset je poskytnutý spoločnosťou FEMTO-ST [60]. Jedná sa o kolekciu vibračných dát priebehu opotrebenia niekoľkých ložísk. Je to verejne dostupný zdroj záznamov ložísk, určený na tvorbu algoritmov schopných predikcie RUL.



Obr. 26: Platforma na kolekciu dát od PRONOSTIA, 1 motor; 2 snímač rýchlosti; 3 prevodovka; 4 meranie krútiaceho momentu; 5 spojka; 6 termočlánok; 7 karta NI DAQ; 8 regulátor tlaku; 9 tlakový valec; 10 snímač sily; 11 testované ložisko; 12 akcelerometer [61].

Zostava sa skladá (vid'. Obr. 26) z rotujúcej časti, ktorá je poháňaná asynchrónnym motorom. Výkon motora je 250 W. Rotujúci pohyb motora je prenášaný na prevodovku, ktorá umožňuje motoru dosiahnuť rýchlosť až 2830 ot/min. Na rotujúcu časť pôsobí radiálna zaťažovacia sila. Časť zostavy s pôsobiacou silou spôsobuje opotrebenie ložiska. Degradácia ložiska je získavaná meracou časťou zostavy. Hodnoty vibrácií sú získané dvomi akcelerometrami umiestnenými 90° od seba. Jeden senzor meria x-os a druhý meria y-os. Vzorkovacia frekvencia akcelerometrov je 25,6 kHz. Každých 10 s sa zaznamená 2560 hodnôt zrýchlenia. Časová doba jedného záznamu z akcelerometra je 0,1 s. Použitý akcelerometer bol DYTRAN 3035B. Senzor mal rozsah merania do ± 50 g. [61]

Všetky použité ložiská mali zhodné rozmery a parametre. Na nižšie uvedenej tabuľke sú vypísané dané rozmery použitých ložísk:

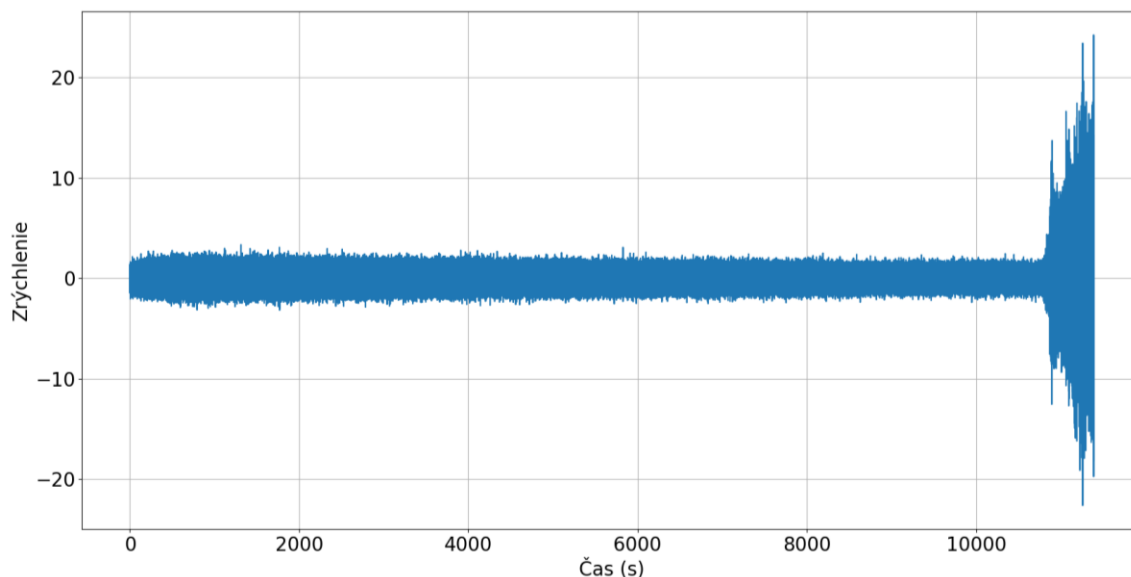
Tab. 4: Základné parametre použitých ložísk [61]

Parameter	Hodnota
Vonkajší priemer D	32 mm
Vnútný priemer d	20 mm
Hrúbka B	7 mm
Statická záťaž	2470 N
Dynamická záťaž	4000 N

Pre každý ASCII súbor boli dáta rozdelené do 6-ch stĺpcov. Prvý stĺpec reprezentuje hodinu konkrétneho záznamu. Ďalšie tri stĺpce vyjadrujú minúty, sekundy a mikro sekundy záznamu. V piatom stĺpci je hodnota z akcelerometru umiestneného horizontálne. V ďalšom stĺpci sa nachádza akcelerometer umiestnený vertikálne.

Tab. 5: Životnosť ložísk

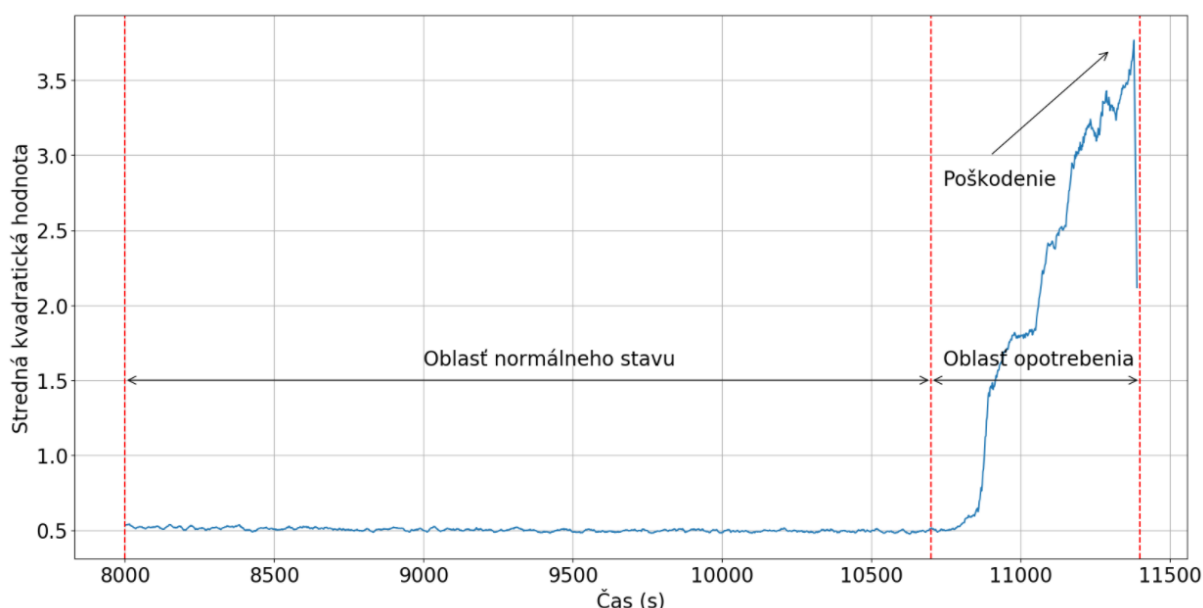
Jednotlivé ložiská	Doba záznamu životnosti ložiska
Ložisko1_1	14 010 s
Ložisko1_2	4 355 s
Ložisko1_3	9 010 s
Ložisko1_4	5 695 s
Ložisko1_5	12 315 s
Ložisko1_6	6 225 s
Ložisko1_7	11 295 s



Obr. 27: Záznam z akcelerometra počas životnosti ložiska

3.5.2 Postup riešenia predikcie RUL ložiska

Zo záznamu priebehu vibrácií boli použité stĺpce s hodnotami vibrácií. Signál sa skladal z dvoch častí. V prvej časti je hodnota z akcelerometru ustálená. V tejto časti je ložisko bez poškodenia a nie je potrebné ho meniť. V druhej časti je možné pozorovať veľký nárast hodnoty kvadratického priemeru. V ložisku nastáva porucha, ktorá sa postupne zväčšuje. Úsek opotrebenia ložiska je dôležitý na identifikáciu poruchy a následné zastavenie operácie obrábacieho stroja. Ložisko je potrebné včas vymeniť, aby nedošlo k vážnejšiemu poškodeniu. Životnosť bola stanovená pri dosiahnutí medznej hranice amplitúdy vibrácií, stanovenej na 20 g [61]. Model predpovedá dobu, ktorá ostáva do dosiahnutia medznej hranice zrýchlenia 20 g.



Obr. 28: Graf znázorňujúci opotrebenie ložiska v priebehu jeho životnosti

Algoritmus na predikciu RUL pozostáva z niekoľkých krokov. Prvým krokom pri učení bolo získať dáta z akcelerometrov. Následne sa signál prefiltraval pomocou funkcie *wavelet*. Záznam zo senzorov obsahoval rušenie, ktoré spôsobovalo menej presnú predikciu RUL. Z dôvodu zefektívnenia učenia a následnej predikcie, bolo zakomponované odstraňovanie šumu. Ďalším krokom v príprave dát bolo získavanie reprezentujúcich parametrov. Zo signálu z časovej oblasti sa získavali štatistické funkcie. Jednou zo štatistických funkcií bol kvadraticky priemer. Kvadratický priemer býva často využívaný ako ukazovateľ stavu súčasti. Výhodnou vlastnosťou je jeho citlivosť na globálne poškodenie ložísk [62]. Následne boli použité funkcie ako stredná absolútna odchýlka, rozdiel vrcholov a koeficient špicatosti. Koeficient špicatosti je charakteristika rozdelenia náhodnej veličiny, ktorá porovnáva dané rozdelenie s normálnym rozdelením pravdepodobnosti [63]. Najprv boli použité všetky 4 štatistické prvky. Na základe porovnania učenia neurónovej siete s koeficientom špicatosti a bez bolo zistené, že neurónová sieť vie presnejšie predikovať RUL bez použitia koeficientu špicatosti. Ďalej bola neurónová sieť trénovaná bez použitia koeficientu špicatosti.

Dáta použité neurónovou sieťou na predikciu RUL boli získané z nižšie uvedených funkcií:

$$x_{rms} = \sqrt{\frac{\sum_{n=1}^N (x_n)^2}{N}} \quad (41)$$

$$x_{mad} = \frac{\sum_{n=1}^N |x_n - x_m|}{N} \quad (42)$$

$$x_{p-p} = \max(x_n) - \min(x_n) \quad (43)$$

Hodnoty získané z extrahovaných funkcií boli ešte zjemnené pomocou funkcie *convolve*. Táto metóda bola zvolená hlavne z dôvodu rýchleho počítania. Konvolúcia je matematický operátor, používaný hlavne v spracovaní signálu. Spojitá konvolúcia je definovaná ako integrál zo súčinu f s otočenou funkciou konvolučného jadra, posunutého do bodu x . Konvolučné jadro je vyjadrené v rovnici (44) ako $g(x)$. [64]

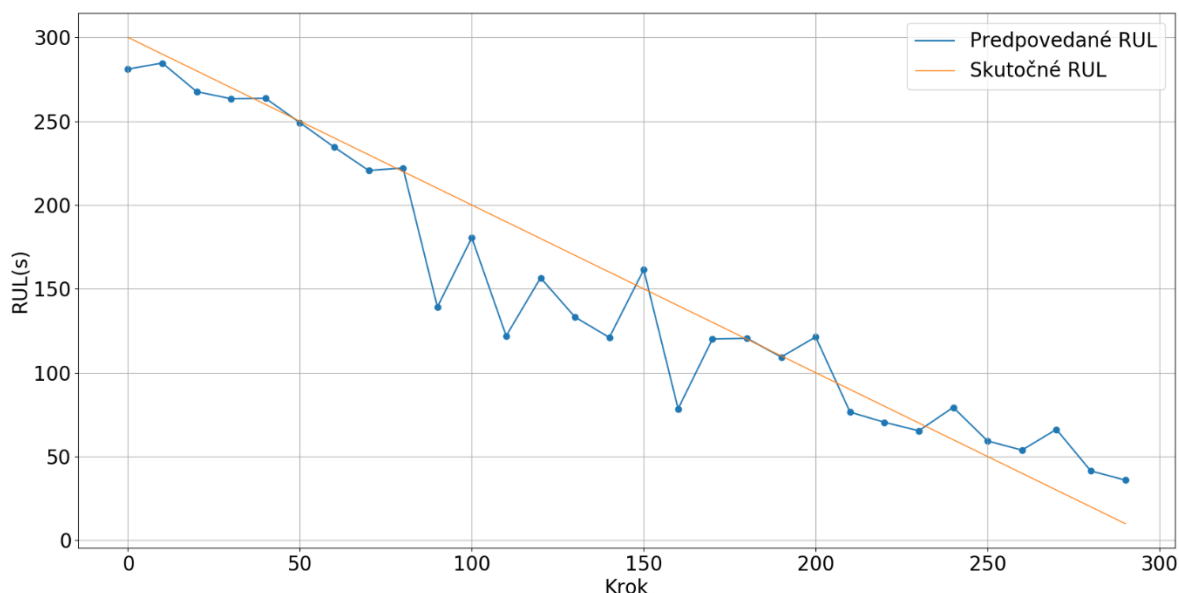
$$(f * g)(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha \quad (44)$$

Nasledujúcim krokom prípravy dát je ich normalizácia. Vďaka normalizácii všetkých vstupných hodnôt na rovnakú veľkosť je neurónová sieť schopná rýchlejšie stanoviť svoje optimálne parametre váh. V tomto prípade vstupné dáta sú hodnoty z funkcií x_{rms} , x_{mad} , x_{p-p} . Na učenie boli použité priebehy zo záznamu vibrácií siedmich ložísk. Vstupné dáta na tréning boli rozdelené na časť učenia a na testovaciu časť. Záznam z vibrácií siedmeho ložiska bol použitý na overenie funkčnosti daného modelu. Pri učení sieť nemala kontakt s týmto záznamom ložiska. Prvýkrát bol záznam siedmeho ložiska sprostredkovaný pre model neurónovej siete počas overenia jeho schopnosti predikcie RUL. Výstupné hodnoty predikcie neurónovej siete boli v sekundách. Rozsah RUL bol určený na 300 sekúnd.

Zvolená neurónová sieť bola rekurentná sieť typu LSTM. Vstupom do siete boli hodnoty z časovej oblasti extrahované prostredníctvom štatistických funkcií. Predikovaná hodnota z neurónovej siete je RUL. Model mal za úlohu predikovať zostávajúci čas ložiska do prekročenia vopred stanovenej kritickej hodnoty zrýchlenia. Časová dĺžka vstupujúceho vektora mala rozmer 5 prvkov. Každý z 5 prvkov na vstupe do neurónovej siete mal rozmer 3 hodnôt. Tieto jednotlivé hodnoty reprezentovali štatistické hodnoty získané z pôvodného signálu, prostredníctvom extrakcií dát uvedených v rovniciach (41), (42) a (43). Neurónová sieť pozostávala zo štyroch vrstiev. Prvé tri vrstvy boli typu LSTM o rozmeroch (120, 180, 180). Štvrtou vrstvou bola plne prepojená vrstva s lineárnou aktivačnou funkciou. Výstup z LSTM vrstvy je propagovaný do plne prepojenej vrstvy neurónovej siete. Pre plne prepojenú sieť bola zvolená lineárna aktivačná funkcia. Na učenie daného modelu bol použitý optimizátor RMSprop. Chyba predikcie modelu sa počítala pomocou funkcie strednej kvadratickej chyby.

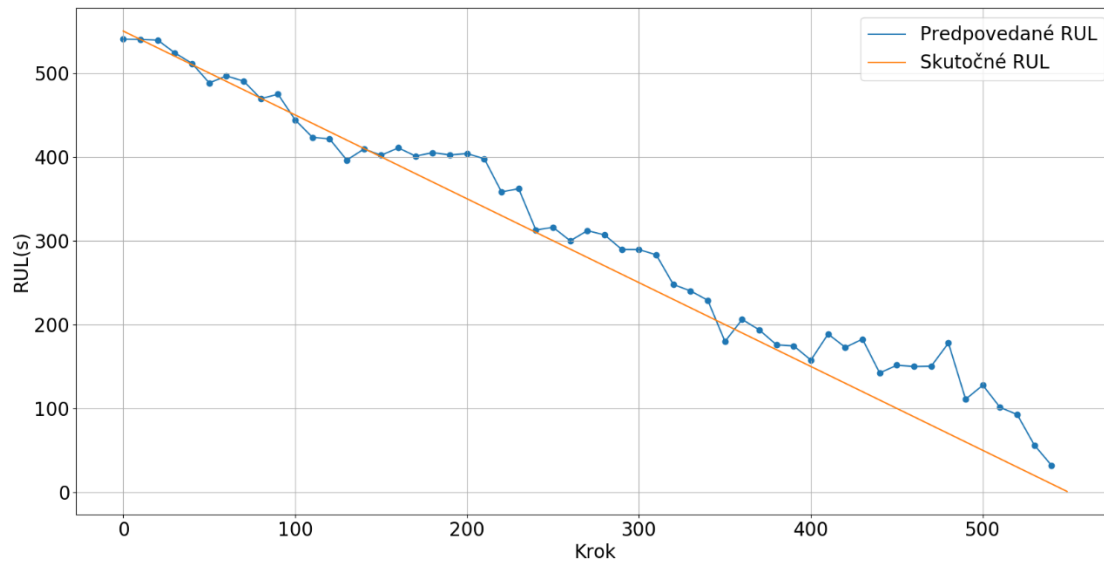
3.5.3 Výsledky

Rekurentná neurónová sieť typu LSTM bola použitá na predikciu zostávajúcej životnosti ložiska. Na základe vstupných hodnôt, ktoré reprezentovali postupnú degradáciu testujúceho ložiska, bol model schopný určiť zostávajúci čas ložiska do dosiahnutia medznej hodnoty zrýchlenia.



Obr. 29: Predikcia zostávajúceho času do poškodenia ložiska

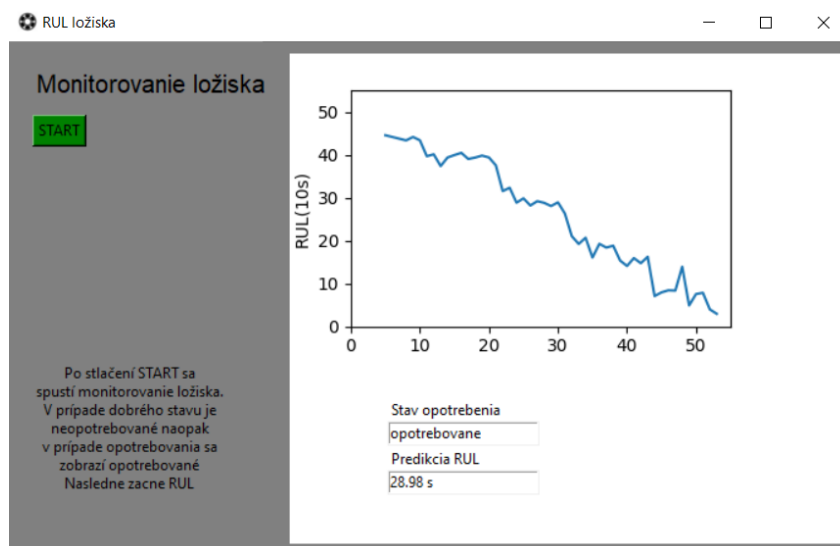
Natrénovaná sieť bola schopná približne určiť predikciu RUL (vid'. Obr. 29: Predikcia zostávajúceho času do poškodenia ložiska). Prvky boli zredukovaným záznamom z 2560 prvkov. Dimenzia prvkov v každom jednom časovom kroku bola o veľkosti 3. Každá z jednotlivých dimenzií reprezentovala jeden štatistický prvok. Doba medzi jednotlivými časovými krokmi je 10 s. Natrénovaná neurónová sieť predikovala stav, každých 10 s. V prípade prvého pokusu je sieť schopná zaznamenať klesajúci trend RUL. Počas testovania algoritmu bolo pozorovaných niekoľko oscilácií predikovaných hodnôt RUL, ktoré znepresňujú výslednú predikciu. Dva zo siedmich záznamov opotrebenia ložiska mali výrazne kratšiu oblasť opotrebenia. Nakoľko dva záznamy ložísk mali nižšiu hodnotu amplitúd kvadratického priemeru, bola zvolená hraničná hodnota vyhovujúca všetkým siedmim ložiskám. Zvolená hraničná hodnota opotrebenia ložiska bola použitá na tréning neurónovej siete. Na optimalizáciu predikcie RUL boli upravené dáta určené na tréning. Dva záznamy ložísk sa zanedbali z dôvodu krátkeho záznamu vibrácií v oblasti opotrebenia (vid'. Obr. 28). V optimalizovanom algoritme je umožnené predikovať dlhšiu dobu zostávajúcej životnosti ložiska. Hodnota predikovaného RUL je 550 s (vid'. Obr. 30).



Obr. 30: Predikcia času do poškodenia ložiska

3.5.4 Aplikácia na monitorovanie ložiska

Pri tvorbe aplikácie na monitorovanie stavu ložiska bol využitý postup z predošlých dvoch kapitol (vid'. kap. 3.4 a 3.5). Aplikácia bola vytvorená pomocou balíčka *tkinter* v Pythone. Cieľom algoritmu bolo monitorovať ložisko prostredníctvom hodnôt získavaných z akcelerometra. Algoritmus sa skladal z dvoch neurónových sietí. Jedna neurónová sieť bola určená na detekciu poruchy ložiska. Druhá neurónová sieť mala za cieľ predikovať ostávajúcu dobu do dosiahnutia hraničnej hodnoty vibrácií. Algoritmus tejto aplikácie je popísaný v predchádzajúcej kapitole (vid'. kap. 3.1.3).



Obr. 31: Program na monitorovanie stavu ložiska

4 Zhodnotenie praktickej časti

Súčasťou práce bolo vytvorenie umelej neurónovej siete na riešenie problémov s monitorovaním strojov v priemysle. Vytvorený algoritmus bol otestovaný na reálnych dátach. Dáta boli získané z verejne dostupných zdrojov. Pôvodne mali byť dáta získané z UVSSR. Žiaľ, kvôli obmedzeniam v súvislosti s pandemiou, nebolo umožnené získať dáta z vyššie uvedeného ústavu. Úlohou práce bolo vytvoriť program na monitorovanie veľkosti opotrebenia frézy a predikciu zostávajúcej životnosti ložiska. Informácie o stave danej súčasti, jeho zostávajúci čas do poruchy bolo následne možné využiť pri plánovaní prediktívnej údržby stroja, s ohľadom na maximálne využitie jeho životnosti a minimalizovanie odstávky stroj. Odstávka stroja v priemysle spôsobuje nemalé ekonomické straty, môže spôsobiť nedodržanie plnenia úloh v požadovanej kvalite, komplexnosti, presnosti, bezchybnosti.

Z dôvodu komplexnosti obrábacích strojov bolo zvolené monitorovanie dvoch najviac namáhaných častí. Vychádzajúc z tohto dôvodu sa vytvoril program na monitorovanie stavu frézy a ložiska. Vybraný typ neurónovej siete v tejto práci bola rekurentná neurónová sieť. Zvolený typ neurónovej siete vie dobre pracovať práve so sekvenčnými dátami. Konkrétne sa použili tri typy neurónových sietí. Na monitorovanie stavu obrábacieho stroja boli použité dáta z troch typov senzorov.

Snahou prvého riešenia bolo vytvoriť algoritmus na monitorovanie stavu opotrebenia frézy. Na testovanie vhodného algoritmu boli použité tri typy rekurentných sietí. Natrénovaná neurónová sieť bola schopná stanoviť mieru opotrebenia frézy na základe vstupných dát zo senzorov. Najpresnejšie výsledky sa získali z neurónovej siete typu GRU. Naopak najhoršou voľbou bola neurónová sieť typu RNN. Po vyhodnotení výsledkov bola najpresnejšia metóda zvolená do aplikácie na nepriame stanovenie opotrebenia ložiska. Nepriama metóda umožňuje online monitorovanie stavu obrábacieho nástroja. Aplikácia mala formu UI a bola vytvorená v Pythone. Jednalo sa o simuláciu monitorovania stavu frézy. V prípade použitia v reálnych podmienkach by bolo potrebné čiastočne upraviť zdrojový kód použitej aplikácie. Model neurónovej siete by mohol byť ďalej využitý na stanovenie fázy opotrebenia nástroja a predikciu jeho poškodenia. Taktiež opotrebením frézy sa stráca požadovaná presnosť obrábania. Mohol by byť vytvorený SW, ktorého úlohou by bolo korigovať hĺbku rezu, na základe informácie o stave opotrebenia frézy.

Druhý problém bol zameraný na detekciu a následnú klasifikáciu poruchy ložiska. Zvolenou neurónovou sieťou na detekciu poruchy bola rekurentná neurónová sieť LSTM. Natrénovaný model bol schopný detegovať poruchu na ložisku. Taktiež bola otestovaná schopnosť modelu určiť o aké poškodenie sa jedná. Vstupom do neurónovej siete boli dáta z akcelerometra. Získané poznatky boli ďalej použité pri vytváraní aplikácie pre monitorovanie stavu ložiska.

Tretí problém riešil predikciu RUL. V prvej časti sa vytvoril algoritmus neurónovej siete na predikciu hodnoty RUL na základe hodnôt z akcelerometra. Predikcia sa uskutočňovala v oblasti zvýšeného opotrebenia ložiska. Cieľom bolo určiť dobu dosiahnutia vopred danej hraničnej hodnoty vibrácií. Model neurónovej siete typu LSTM bol po optimalizácii schopný pomerne presne určiť zostávajúcu životnosť ložiska v danej oblasti. Konkrétny model sa použil v aplikácii na monitorovanie stavu ložiska. Aplikácia sa skladá z dvoch neurónových sietí. Úlohou prvej neurónovej siete je detegovať začiatok zvýšeného opotrebenia ložiska. Po detekcii opotrebenia sa spustí druhá časť aplikácie. V tejto časti sa využíva neurónová sieť na predikciu RUL ložiska. Aplikácia bola vytvorená v Pythone za použitia balíčka Tkinter.

5 Záver

Cieľom práce bolo zoznámenie sa s umelými neurónovými sieťami a ich použitím v priemysle. Súčasťou tejto práce bolo vytvorenie programu založeného na neurónových sieťach, ktorý by bol schopný monitorovať stav najviac namáhaných komponentov obrábacieho stroja.

Program bol vytvorený v jazyku Pythone za použitia hlavne knižnice Keras. Použitá platforma na programovanie a prácu s dátami bola Jupyter Notebook. Dáta na trénovanie navrhnutého modelu neurónovej siete boli z verejne dostupných zdrojov. Práca je členená na dva hlavné celky.

V prvej časti práce je rozobratý teoretický základ, potrebný na riešenie praktickej časti. Taktiež sa teoretická časť tejto práce venuje rešerši iných prác, ktoré riešili problém monitorovania priemyselných strojov.

Druhá časť je venovaná riešeniu troch problémov. Riešenie daných problémov malo za cieľ vytvoriť komplexné monitorovanie obrábacieho stroja. Konkrétne bolo riešené monitorovanie opotrebenia frézy a predikcia RUL ložiska.

Prvý riešený problém praktickej časti bolo stanovenie opotrebenia frézy. Použité boli tri typy rekurentných neurónových sietí, ktoré boli následne porovnané. Najlepšou voľbou bola neurónová sieť typu GRU. Vybratá neurónová sieť mala za cieľ stanoviť zo senzorových dát stav opotrebenia obrábacieho nástroja. Pre možné uplatnenie natrénovaného neurónového modelu bola vytvorená aplikácia UI s použitím knižnice Tkinter. Nakoľko poskytnuté dáta na trénovanie neurónovej siete pozostávali len z troch záznamov meraní, v budúcnosti by bolo vhodné použiť viac dát na trénovanie neurónovej siete a testovanie prezentovaného algoritmu kvôli zvýšeniu presnosti stanovenia miery opotrebenia frézy.

Ďalším riešeným problémom bola detekcia poruchy ložiska a následná klasifikácia danej poruchy. Zvolená neurónová sieť typu LSTM detegovala a klasifikovala poruchu vyskytujúcu sa na monitorovanom ložisku. Dáta používané na monitorovanie boli získavané z akcelerometra. Neurónová sieť bola schopná relatívne presne klasifikovať monitorovaný záznam z vibrácie ložiska (*vid'. Obr. 25*).

V treťom probléme sa riešila predikcia zostávajúcej životnosti ložiska. Použitý typ neurónovej siete bol LSTM. Neurónová sieť mala za cieľ z extrahovaných dát z akcelerometra určiť hodnotu RUL. Následne, po natrénovaní modelu, bola vytvorená aplikácia, využívajúca dve neurónové siete. Jedna neurónová sieť bola určená na detekciu poruchy a druhá bola určená na predikciu RUL.

V tejto, práci na základe rešerše jednotlivých algoritmov neurónovej siete, boli vytvorené spomenuté tri algoritmy na monitorovanie strojov v priemysle. Jednotlivé algoritmy boli schopné monitorovať požadované komponenty stroja.

Prínosom tejto práce bolo otestovanie vhodnosti umelých neurónových sietí typu LSTM, RNN a GRU na monitorovanie obrábacích strojov. Taktiež by sa algoritmy mohli po úprave použiť na online monitorovanie. V budúcnosti by bolo vhodné otestovať jednotlivé algoritmy na online monitorovaní daných komponentov.

6 Zoznam použitých zdrojov

- [1] MOHAJAN, Haradhan. *The Second Industrial Revolution has Brought Modern Social and Economic Developments* [online]. 2020 [vid. 2021-04-09]. Dostupné z: https://www.researchgate.net/publication/338670501_The_Second_Industrial_Revolution_has_Brought_Modern_Social_and_Economic_Developments
- [2] Základné prvky Industry 4.0, Digitalizácia, Smart Factory. *Industry 4.0* [online]. [vid. 2021-04-09]. Dostupné z: <https://industry4.sk/o-industry-4-0/principy/>
- [3] IRVAN, Cerita. *INDUSTRY 4.0: The Digital Technology Transformation* [online]. 2018 [vid. 2021-05-15]. Dostupné z: <https://bunzut.wordpress.com/2018/05/25/industry-4-0-the-digital-technology-transformation/>
- [4] AZARMIPOUR, Mahyar, Haitham ELFAHAM, Caspar GRIES a Ulrich EPPLÉ. PLC 4.0: A Control System for Industry 4.0. In: *IECON Proceedings (Industrial Electronics Conference)* [online]. B.m.: IEEE Computer Society, 2019, s. 5513–5518 [vid. 2021-04-09]. ISBN 9781728148786. Dostupné z: <https://ieeexplore.ieee.org/document/8927026>
- [5] Artificial intelligence. *Wikipedia* [online]. 2021 [vid. 2021-04-09]. Dostupné z: https://en.wikipedia.org/wiki/Artificial_intelligence
- [6] MITCHELL, Tom. *Machine Learning* [online]. New York: McGraw Hill, 1997 [vid. 2021-04-09]. ISBN 0-07-042807-7. Dostupné z: <http://www.cs.cmu.edu/~tom/mlbook.html>
- [7] HINTON, Geoffrey. *CSC321: Neural Networks Lecture 13: Learning without a teacher: Autoencoders and Principal Components Analysis* [online]. [vid. 2021-05-16]. Dostupné z: <https://www.cs.toronto.edu/~hinton/csc321/notes/lec13.pdf>
- [8] Deep learning. *Wikipedia* [online]. 2021 [vid. 2021-04-09]. Dostupné z: https://en.wikipedia.org/wiki/Deep_learning
- [9] CHENG, Lixue, Nikola B. KOVACHKI, Matthew WELBORN a Thomas F. MILLER. Regression-clustering for Improved Accuracy and Training Cost with Molecular-Orbital-Based Machine Learning. *Journal of Chemical Theory and Computation* [online]. 2019 [vid. 2021-05-16]. Dostupné z: <http://arxiv.org/abs/1909.02041>
- [10] TANG, Gang, Youguang ZHOU, Huaqing WANG a Guozheng LI. Prediction of bearing performance degradation with bottleneck feature based on LSTM network. *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* [online]. 2018 [vid. 2021-04-09]. Dostupné z: <https://ieeexplore.ieee.org/document/8409564/>
- [11] YUAN, Mei, Wu YUTING a Lin LI. *Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network* [online]. 2016 [vid. 2021-03-05]. Dostupné z: <https://ieeexplore.ieee.org/document/7748035>
- [12] LIU, Ruonan, Guotao MENG, Boyuan YANG, Chuang SUN a Xuefeng CHEN. Dislocated Time Series Convolutional Neural Architecture: An Intelligent Fault Diagnosis Approach for Electric Machine. *IEEE Transactions on Industrial Informatics* [online]. 2017, **13**(3), 1310–1320 [vid. 2021-05-16]. ISSN 15513203. Dostupné z: <https://ieeexplore.ieee.org/document/7797508>
- [13] SHAO, Y. a K. NEZU. Prognosis of remaining bearing life using neural networks. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* [online]. 2000 [vid. 2021-04-08]. Dostupné z: <https://journals.sagepub.com/doi/10.1243/0959651001540582>
- [14] YU, Guangbin, Shi ZHUO, Jun YU a Ke LIU. Remaining useful life prediction of rolling bearings using InfoLSGAN and AC algorithm. *Hangkong Dongli Xuebao/Journal of Aerospace Power* [online]. 2020 [vid. 2021-04-07]. Dostupné z: <https://www.hindawi.com/journals/mpe/2020/2153235/>

- [15] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* [online]. 1958, **65**(6), 386–408 [vid. 2021-05-15]. ISSN 1939-1471. Dostupné z: <http://doi.apa.org/getdoi.cfm?doi=10.1037/h0042519>
- [16] ELMAN, Jeffrey. *Finding Structure in Time* [online]. 1990 [vid. 2021-04-09]. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/036402139090002E>
- [17] BOURLARD, H. a Y. KAMP. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics* [online]. 1988, **59**(4–5), 291–294 [vid. 2021-04-09]. ISSN 0340-1200. Dostupné z: <http://link.springer.com/10.1007/BF00332918>
- [18] BENGIO, Yoshua, Pascal LAMBLIN, Dan POPOVICI a Hugo LAROCHELLE. Greedy Layer-Wise Training of Deep Networks. In: *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference* [online]. B.m.: The MIT Press, 2007 [vid. 2021-04-09]. Dostupné z: <http://www.iro.umontreal.ca/~lisa/pointeurs/BengioNips2006All.pdf>
- [19] LIETAVCOVÁ, Zuzana. *KONVOLUČNÍ NEURONOVÉ SÍTĚ* [online]. B.m., 2018 [vid. 2021-04-03]. VUT. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/114593>
- [20] TOH, Gyungmin a Junhong PARK. Review of Vibration-Based Structural Health Monitoring Using Deep Learning. *Applied Sciences* [online]. 2020, **10**(5), 1680 [vid. 2021-05-15]. ISSN 2076-3417. Dostupné z: <https://www.mdpi.com/2076-3417/10/5/1680>
- [21] KARPATY, Andrej, Justin JOHNSON a Li FEI-FEI. Visualizing and Understanding Recurrent Networks. *Cornell University* [online]. 2015 [vid. 2021-04-05]. Dostupné z: <http://arxiv.org/abs/1506.02078>
- [22] SCHMIDHUBER, Jürgen. Deep Learning in neural networks: An overview. *ScienceDirect* [online]. 2015 [vid. 2021-04-01]. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0893608014002135?via%3Dihub>
- [23] SHERSTINSKY, Alex. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *ScienceDirect* [online]. 2020 [vid. 2021-02-05]. Dostupné z: <https://doi.org/10.1016/j.physd.2019.132306>
- [24] HOSHI, Ikuo, Tomoyoshi SHIMOBABA, Takashi KAKUE a Tomoyoshi ITO. Single-pixel imaging using a recurrent neural network combined with convolutional layers. *Optics Express* [online]. 2020, **28**(23), 34069 [vid. 2021-05-15]. ISSN 1094-4087. Dostupné z: <https://doi.org/10.1364/OE.410191>
- [25] ORJEŠEK, Richard. Rekurentné neurónové siete. *Umela inteligencia* [online]. [vid. 2021-04-10]. Dostupné z: <https://umelainteligencia.sk/rekurentne-neuronove-siete/>
- [26] ZHAO, Rui, Ruqiang YAN, Zhenghua CHEN, Kezhi MAO, Peng WANG a Robert X. GAO. Deep learning and its applications to machine health monitoring. *ScienceDirect* [online]. 2019 [vid. 2021-03-02]. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0888327018303108>
- [27] *Rekurentné neurónové siete* [online]. [vid. 2021-04-10]. Dostupné z: http://www2.fiit.stuba.sk/~cernans/nn/nn_texts/neuronove_siete_priesvitky_0708_Q.pdf
- [28] CHUNG, Junyoung, Caglar GULCEHRE, KyungHyun CHO a Yoshua BENGIO. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *Cornell University* [online]. 2014 [vid. 2021-02-12]. Dostupné z: <http://arxiv.org/abs/1412.3555>
- [29] Understanding LSTM Networks. In: *Colah's blog* [online]. 2015 [vid. 2021-05-16]. Dostupné z: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

- [30] HEIDARYSAFA, Mojtaba, Kamran KOWSARI, Laura BARNES a Donald BROWN. Analysis of Railway Accidents' Narratives Using Deep Learning. *ResearchGate* [online]. 2018 [vid. 2021-05-05]. Dostupné z: https://www.researchgate.net/publication/328353038_Analysis_of_Railway_Accidents'_Narratives_Using_Deep_Learning
- [31] SZANDAŁA, Tomasz. *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks* [online]. nedatováno [vid. 2021-04-16]. Dostupné z: https://www.researchgate.net/publication/344757203_Review_and_Comparison_of_Commonly_Used_Activation_Functions_for_Deep_Neural_Networks
- [32] A list of cost functions used in neural networks, alongside applications. In: *Stackexchange* [online]. 2021 [vid. 2021-04-09]. Dostupné z: <https://stats.stackexchange.com/questions/154879/a-list-of-cost-functions-used-in-neural-networks-alongside-applications>
- [33] Regresní analýza. In: *Wikipedia* [online]. 2020 [vid. 2021-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Regresní_analýza
- [34] GROVER, Prince. 5 Regression Loss Functions All Machine Learners Should Know. *HeartBeat* [online]. 2018 [vid. 2021-04-09]. Dostupné z: <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>
- [35] CAVAZZA, Jacopo a Vittorio MURINO. Active Regression with Adaptive Huber Loss. *Cornell University* [online]. 2016 [vid. 2021-04-09]. Dostupné z: <http://arxiv.org/abs/1606.01568>
- [36] ZHANG, Zhilu a Mert R SABUNCU. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *Cornell University* [online]. 2018 [vid. 2021-04-09]. Dostupné z: <https://papers.nips.cc/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf>
- [37] GÓMEZ, Raúl. Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names. *Gombru* [online]. 2018 [vid. 2021-04-16]. Dostupné z: https://gombru.github.io/2018/05/23/cross_entropy_loss/
- [38] GROSSE, Roger. *Lecture 8: Optimization* [online]. [vid. 2021-04-09]. Dostupné z: https://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/readings/L08_Optimization.pdf
- [39] WILSON, D.R. a T.R. MARTINEZ. The need for small learning rates on large problems. *IEEE* [online]. 2001 [vid. 2021-05-21]. Dostupné z: <http://ieeexplore.ieee.org/document/939002/>
- [40] LI, Mu, Tong ZHANG, Yuqiang CHEN a Alexander J SMOLA. Efficient Mini-batch Training for Stochastic Optimization. *ResearchGate* [online]. 2014 [vid. 2021-04-16]. Dostupné z: <http://dx.doi.org/10.1145/2623330.2623612>
- [41] RUDER, Sebastian. An overview of gradient descent optimization algorithms. *Cornell University* [online]. 2016 [vid. 2021-05-01]. Dostupné z: <https://arxiv.org/abs/1609.04747>
- [42] PARKS, David F, Manfred K WARMUTH, Shawfeng DONG, J Xavier PROCHASKA a Quentin WILLIAMS. A study of the Exponentiated Gradient +/-algorithm for stochastic optimization of neural networks. *eScholarship* [online]. 2019 [vid. 2021-04-16]. Dostupné z: <https://escholarship.org/uc/item/4ck5k544>
- [43] OOOYEN, A a B NIENHUIS. *Improving the Convergence of the Black-Propagation Algorithm* [online]. 1991 [vid. 2021-04-09]. Dostupné z: <https://anvooyen.home.xs4all.nl/papers/backprop.pdf>
- [44] *Derivation of Backpropagation* [online]. [vid. 2021-04-17]. Dostupné z: <https://www.cs.swarthmore.edu/~meeden/cs81/s10/BackPropDeriv.pdf>

- [45] RUMELHART, David, G HINTON a R WILLIAMS. *Learning representations by back-propagating errors* [online]. 1986 [vid. 2021-05-17]. Dostupné z: https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf
- [46] ALI, Peshawa a Rezhna FARAJ. Data Normalization and Standardization: A Technical Report. *ResearchGate* [online]. 2014 [vid. 2021-05-07]. Dostupné z: https://www.researchgate.net/publication/340579135_Data_Normalization_and_Standardization_A_Technical_Report
- [47] RAD, Javad Soltani, Youmin ZHANG a Chevy CHEN. A novel local time-frequency domain feature extraction method for tool condition monitoring using s-transform and genetic algorithm. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* [online]. B.m.: IFAC Secretariat, 2014, s. 3516–3521 [vid. 2021-04-17]. ISBN 9783902823625. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S147466701642149X>
- [48] ZHANG, Cunji, Xifan YAO, Jianming ZHANG a Hong JIN. Tool Condition Monitoring and Remaining Useful Life Prognostic Based on a Wireless Sensor in Dry Milling Operations. *Sensors* [online]. 2016, **16**(6), 795 [vid. 2021-05-19]. ISSN 1424-8220. Dostupné z: <http://www.mdpi.com/1424-8220/16/6/795>
- [49] GRAPS, Amara. *Lab 8. Introduction to Wavelets* [online]. nedatováno [vid. 2021-04-17]. Dostupné z: <https://www.eecis.udel.edu/~amer/CISC651/IEEEwavelet.pdf>
- [50] BARU, Aditya. Three Ways to Estimate Remaining Useful Life Predictive Maintenance with MATLAB. *Mathworks* [online]. 2018 [vid. 2021-03-17]. Dostupné z: <https://nl.mathworks.com/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance.html>
- [51] LI, Guofa, Yanbo WANG, Jialong HE, Qingbo HAO, Haiji YANG a Jingfeng WEI. Tool wear state recognition based on gradient boosting decision tree and hybrid classification RBM. *Springer* [online]. 2020 [vid. 2021-04-18]. Dostupné z: <http://link.springer.com/10.1007/s00170-020-05890-x>
- [52] LIU, Hui, Zhenyu LIU, Weiqiang JIA, Xianke LIN a Shuo ZHANG. A novel transformer-based neural network model for tool wear estimation. *ResearchGate* [online]. 2020 [vid. 2021-04-25]. Dostupné z: https://www.researchgate.net/publication/339006617_A_novel_transformer-based_neural_network_model_for_tool_wear_estimation/citation/download
- [53] LI, X., B. S. LIM, J. H. ZHOU, S. HUANG, S. J. PHUA, K. C. SHAW a M. J. ER. Fuzzy neural network modelling for tool wear estimation in dry milling operation. *Annual Conference of the Prognostics and Health Management Society, PHM 2009* [online]. 2009, 1–11 [vid. 2021-04-26]. Dostupné z: <https://papers.phmsociety.org/index.php/phmconf/article/view/1403>
- [54] MOHANRAJ, T., S. SHANKAR, R. RAJASEKAR, N.R. SAKTHIVEL a A. PRAMANIK. Tool condition monitoring techniques in milling process — a review. *Journal of Materials Research and Technology* [online]. 2020, **9**(1), 1032–1042 [vid. 2021-05-15]. ISSN 22387854. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S2238785418313061>
- [55] GOEBEL, Kai, Nasa AMES, Alice AGOGINO a U C BERKELEY. Documentation for Mill Data Set [online]. 1996, 1–10 [vid. 2021-03-19]. Dostupné z: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>
- [56] TOBON-MEJIA, Diego, Kamal MEDJAHHER a Nouredine ZERHOUNI. CNC Machine Tool 's wear diagnostic and prognostic by using dynamic bayesian networks. *Elsevier* [online]. 2012 [vid. 2021-03-15]. Dostupné z: <https://hal.archives-ouvertes.fr/hal-00672204>
- [57] Download a Data File. *Case Western Reserve University* [online]. [vid. 2021-05-17]. Dostupné z: <https://csegroups.case.edu/bearingdatacenter/pages/download-data-file>
- [58] What Are Bearings? Let's learn about the basic functions of bearings! *Koyo* [online]. [vid. 2021-04-11]. Dostupné z: <https://koyo.jtekt.co.jp/en/2019/06/column01-01.html>

- [59] *ROLLING CONTACT BEARINGS Classification, design aspects and selection* [online]. [vid. 2021-04-11]. Dostupné z: <https://www.bcrec.net.in/ME/design/15.pdf>
- [60] NECTOUX, P, R GOURIVEAU, K MEDJAHHER, E RAMASSO, B CHEBEL-MORELLO, N ZERHOUNI a C VARNIER. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In: *In IEEE International Conference on Prognostics and Health Management*. [online] 2012 [vid. 2021-04-19] Dostupné z: <https://www.femto-st.fr/en/Research-departments/AS2M/Research-groups/PHM>
- [61] IEEE PHM 2012 Prognostic challenge - Outline , Experiments , Scoring of results , Winners. *SCRIBD* [online]. 2012 [vid. 2021-03-19]. Dostupné z: <https://www.scribd.com/document/344094074/IEEEPHM2012-Challenge-Details-Results-and-Winners>
- [62] HANLY, Steve. Vibration Measurements: Vibration Analysis Basics. *Endaq blog* [online]. 2018 [vid. 2021-04-19]. Dostupné z: <https://blog.endaq.com/vibration-measurements-vibration-analysis-basics>
- [63] Koeficient špicatosti. *Wikipedia* [online]. 2014 [vid. 2021-04-16]. Dostupné z: https://sk.wikipedia.org/wiki/Koeficient_špicatosti
- [64] Konvoluce. *Wikipedia* [online]. 2021 [vid. 2021-05-10]. Dostupné z: <https://cs.wikipedia.org/wiki/Konvoluce>

7 Zoznam použitých symbolov

$W[-]$	– matica váh vstupných hodnôt
$x_t[-]$	– vstupné hodnoty v čase t
$b[-]$	– bias
$h_t[-]$	– vnútorný stav siete v čase t
$o^t[-]$	– výstupná brána
$c_t[-]$	– kanál
$f_t[-]$	– zabúdacia brána
$i_t[-]$	– vstupná brána
$W_c, V_c[-]$	– matica váh vnútorného kanálu
$b_c[-]$	– bias kanálu
$W_f, V_f[-]$	– matica váh zabúdacej brány
$b_f[-]$	– bias zabúdacej brány
$W_i, V_i[-]$	– matica váh vstupnej brány
$b_i[-]$	– bias vstupnej brány
$W_o, V_o[-]$	– matica váh výstupnej brány
$b_o[-]$	– bias výstupnej brány
$z_t[-]$	– obnovovacia brána
$W_z, U_z[-]$	– matica váh brány obnovenia
$r_t[-]$	– resetovacia brána
$W_r, U_r[-]$	– matica váh resetovacej brány
$y_i[-]$	– skutočná hodnota na výstupe z neurónovej siete
$\hat{y}_i[-]$	– hodnota predikcie
$HU[-]$	– Hubertova metóda
$H[-]$	– entropia
$p(y_i)[-]$	– pravdepodobnosť
$\alpha[-]$	– koeficient učenia
$\nabla J(w)[-]$	– Jacobiho matica
$v_t[-]$	– vektor na úpravu váh neurónovej siete
$\gamma[-]$	– priepustnosť hodnôt z predošlého vektora
$G_t[-]$	– diagonálna matica, ktorej diagonálny prvok je sumou druhej mocniny gradientov
$\varepsilon[-]$	– vyrovnávací hodnota
$E[g^2]_t[-]$	– priemerná hodnota druhého gradientu
$C_{cel}[-]$	– celková chyba neurónovej siete na výstupe
$y_{y1}[-]$	– výstup z poslednej vrstvy
$net_{y1}[-]$	– suma všetkých váh na výstupnej vrstve neurónovej siete
$y_{s1}[-]$	– výstup zo skrytej vrstvy
$net_{s1}[-]$	– suma všetkých váh pre jeden neurón v skrytej vrstve neurónovej siete
$t_f[s]$	– čas poruchy
$t_s[s]$	– čas začiatku predikcie
$x_{rms}[-]$	– kvadratický priemer
$x_{mad}[-]$	– stredná absolútna odchýlka
$x_{p-p}[-]$	– rozdiel minima a maxima

8 Zoznam použitých skratiek

AI (*Artificial Intelligence*) – umelá inteligencia
ML (*Machine Learning*) – strojové učenie
DL (*Deep Learning*) – hĺbkové učenie
RNN (*Recurent Neural Network*) – rekurentná neurónová sieť
LSTM (*Long Short – Term Memory*) – dlhodobá-krátkodobá pamäť, typ rekurentnej siete
GRU (*Gated Recurent Neural Network*) – typ rekurentnej neurónovej siete
FFNN (*Feedforward Neural Network*) – dopredná neurónová sieť
AE (*Auto Encoder*) – auto enkodér
DBN (*Deep Belief Network*) – sieť hĺbkového presvedčenia
CNN (*Convolutional Neural Network*) – konvolučná neurónová sieť
MSE (*Mean Square Error*) – kvadratická hodnota chyby
MAE (*Mean Absolute Error*) – absolútna hodnota chyby
RUL (*Remaining Usefull Life*) – odhad zostávajúceho života
CNC (*Computer numerical control*) – číslicové riadenie
GD (*Gradient Descent*) – gradientný pokles
SGD (*Stochastic Gradient Descent*) – stochastický gradientný pokles
RMSprop (*Root Mean Square Propagation*) – stredná kvadratická propagácia

9 Zoznam použitých obrázkov

Obr. 1: Evolúcia priemyslu[3]	13
Obr. 2: Rozbalenie rekurentnej neurónovej siete [24].....	17
Obr. 3: Rozdelenie RNN podľa pomeru vstupov a výstupov z neurónovej siete	18
Obr. 4: Schematické porovnanie architektúry GRU a LSTM [30].....	20
Obr. 5: Porovnanie dvoch hodnôt koeficientov učenia a ich vplyv na konvergenciu	24
Obr. 6: Diagram spätnej propagácie	27
Obr. 7: Vykreslenie Daubechovej wavelet funkcie	30
Obr. 8: Schéma procesu filtrácie pre wavelet funkcie v balíčku pywt.....	30
Obr. 9: Znázornenie Threshold metódy na určenie RUL	31
Obr. 10: Schéma trénovania a testovania neurónovej siete	32
Obr. 11: Algoritmus na stanovenie stavu frézy	33
Obr. 12: Klasifikácia stavu monitorovaného ložiska.....	34
Obr. 13: Schéma algoritmu na monitorovanie a predikciu stavu ložiska.....	35
Obr. 14: Schéma merania dát na predikciu opotrebenia frézy[55].....	37
Obr. 15: Porovnanie signálu z dynamometra pre 1. meranie a pre 315. meranie.....	38
Obr. 16: Porovnanie signálu z akcelerometra pre 1. meranie a 315. meranie	38
Obr. 17: Porovnanie signálu z akustického senzora pre 1. a 315. meranie	38
Obr. 18: Graf opotrebenia frézy rozdelený do troch sekcií	39
Obr. 19: Porovnanie troch meraní opotrebenia	39
Obr. 20: Predikcia stavu frézy prostredníctvom RNN	41
Obr. 21: Predikcia stavu frézy prostredníctvom LSTM	41
Obr. 22: Predikcia stavu frézy prostredníctvom GRU	42
Obr. 23: Aplikácia na monitorovanie stavu frézy	42
Obr. 24: Priebeh trénovania neurónovej siete	44
Obr. 25: Chybová matica pre detekciu poruchy ložiska.....	44
Obr. 26: Platforma na kolekciu dát od PRONOSTIA [54].....	45
Obr. 27: Záznam z akcelerometra počas života ložiska	46
Obr. 28: Graf znázorňujúci opotrebenie ložiska v priebehu jeho životnosti	47
Obr. 29: Predikcia zostávajúceho času do poškodenia ložiska	49
Obr. 30: Predikcia času do poškodenia ložiska	50
Obr. 31: Program na monitorovanie stavu ložiska	50

10 Zoznam použitých tabuliek

Tab. 1: Porovnanie výhod a nevýhod jednotlivých architektúr neurónových sietí [20] .	16
Tab. 2: Hodnoty základných parametrov obrábacieho stroja[53]	37
Tab. 3: Výsledky MSE pre použité neurónové siete.....	40
Tab. 4: Základné parametre použitých ložísk[54]	46
Tab. 5: Životnosť ložísk.....	46